Disk Operating System

Command Reference and Error Messages

Version 6

IBM

Programming Family

Disk Operating System

Command Reference and Error Messages

Version 6

IBM
Programming Family

First Edition (June 1993)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Requests for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

Contents

About This Book

Welcome to DOS, the most widely used operating system for personal computers. DOS includes many new features that turn your computer into a powerful tool for your business or personal use.

If DOS is not yet set up on your computer system, you will need to use the Setup program.

Who Should Read this Book

This guide is a comprehensive reference for DOS commands and error messages. The information in this book can be used by people who have little experience with computer systems or by those who are more familiar with computer systems and DOS.

Prerequisite Publications

This book is not written for novice computer users. Anyone new to DOS or anyone who is unfamiliar with personal computers should refer to the *Everyday DOS for PC DOS 6* book.

Related Documentation

The IBM DOS library includes the following publications:

- *PC DOS Installation Guide*

  This book gives you the instructions you need to set up DOS on your computer.

- *Everyday DOS for PC DOS 6*

  This book gives the beginning user basic information to become productive quickly. It also has information on 20 frequently used DOS commands. You will find step-by-step procedures for using DOS Shell and information about customizing DOS Shell in this book. Everyday DOS for PC DOS 6 can be optionally purchased. See the coupons included with this book.

- *PC DOS Keyboards and Code Pages*

  This book, an optional purchase item, contains examples of keyboard layouts, code page tables, and available accented characters that can be used with DOS.

- *DOS 5.02 Technical Reference*

  This book, an optional purchase item, is written for programmers who develop applications for IBM Personal Computers and Personal System/2*

- *PC DOS User's Guide*

  This book contains step-by-step procedures for experienced DOS and computer users. It also contains information about DOS optional tools.

- Help for DOS

  To get a list of commands for which there is online help available, type the word `help` at the DOS command prompt. To get help on a particular command in the list, type `help` followed by the command name or the command name followed by /?.

The online help gives brief information about each command, instructions on the format to use when typing this command, brief explanations about each switch, and notes to assist you.

How this Book is Organized

This book is divided into the following two parts:

- Part 1. "Command Reference" provides basic information about command-line, CONFIG.SYS, and menu configuration commands and DOS operations. It gives an alphabetical reference with complete descriptions of the commands used in DOS.

  It also provides a chapter on device drivers and how to edit specific profile (.INI) files.

- Part 2. "Error Messages" Provides error message information. The error messages are arranged in alphabetic order. The information includes a description of the error, the cause, and a suggestion for correcting the problem.

New Commands for DOS

With the enhancement of DOS and the addition of many new features, more commands have been added to DOS.

The new commands are:

Command	**Purpose**
@	Placed in front of a command in your batch program and prevents the single command from being displayed.
;	Specifies that the current line is a descriptive comment and should not be carried out. Insert this character at the beginning of a line in your CONFIG.SYS file.
?	Specifies that DOS is to ask for confirmation before carrying out the current command. This command can be used only with certain CONFIG.SYS commands.
CHOICE	Prompts the user to make a choice in a batch program. Displays a specified prompt and pauses for the user to choose from among a specified set of keys. You can use this command only in batch programs.
CPBACKUP	Starts a full-screen utility program that is used to back up one or more files from one disk to another. **Note:** This command replaces the BACKUP command.
CPDIR	Starts the CPDIR program. This program determines the number of diskettes and the correct order of a high-speed or medium-speed diskette backup, as well as provides information about how the backup was made.

CPSCHED
Starts the terminate-and-stay-resident CPSCHED program that is used to schedule other DOS programs to run at a preset date and time.

DATAMON
Starts the terminate-and-stay-resident DATAMON program that is used to guard against data loss on your system.

DEFRAG
Reorganizes the files on a disk to optimize disk performance. Do not use this command when you are running Windows.

DELTREE
Deletes a directory and all the subdirectories and files in it.

E
Starts the DOS full-screen editor that enables you to create, edit, save, and print ASCII text.

IBMAVD
Checks your system for computer viruses.

IBMAVDR
Starts an automatic anti-virus scan at system startup.

IBMAVDW
Starts the Windows anti-virus program.

IBMASH
Starts the anti-virus shield program.

IBMAVSP
Starts the AntiVirus stand-alone program that is used for recovery from castastrophic virus damage.

INSTALLHIGH
Loads a terminate-and-stay-resident program into upper memory blocks.

MOVE
Moves files and renames files and directories.

NUMLOCK
Specifies whether the NUMLOCK setting of the numeric keypad is initially set to ON or OFF.

PCFORMAT
Used to format PCMCIA** static memory cards.

PCMCS
Loads PCMCIA Card Service 2.0 support.

PCMFDD
Emulates diskette drives from PCMCIA cards.

PCMINFO
Provides information about each PCMCIA socket and the card inserted in it.

PCMMTD
Installs one or more Memory Technology Drivers (MTDs) so that PCMCIA cards with different memory technologies can be supported.

PCMSCD
Installs one or more Super Client Drivers (SCDs) so that PCMCIA device cards (such as fax and modem) can be supported.

PENDOS
Loads the files necessary to support pen-based systems.

QCONFIG
Displays detailed technical information about your system.

RAMSETUP
Starts the RAMBoost setup program that enables you to customize the way RAMBOOST uses the upper memory blocks (UMB) of your computer.

SCHEDULE

Starts the Schedule program that enables you to specify a preset date and time for a DOS program to run.

Note: The CPSCHED program must be loaded before Schedule can work.

SMARTDRV

Creates a disk cache in extended or expanded memory.

WNBACKUP

This command works only with Windows Version 3.1. It initiates the making of a backup copy of data to diskettes, tape, or a network drive.

WNSCHEDL

Starts the scheduler program for Windows (version 3.1 only) that enables you to specify a time and date for an application program to be started, or for a document to be opened.

WNUNDEL

Starts the undelete program for Windows (version 3.1 only) that recovers erased files.

WPCMINFO

Starts the program for Windows (version 3.1 only) that displays the information about PCMCIA sockets and cards on the host system.

Enhanced Commands for DOS

The following commands have been enhanced for this version of DOS.

COMMAND
DEVICEHIGH
DRVLOCK
EMM386
FDISK
KEYB
LOADHIGH
MEM
SETUP (only for installation)
SETVER
UNDELETE
UNFORMAT

New Device Drivers for DOS

Following are the device drivers that have been added for DOS.

COUNTRY.SYS
INTERLNK.EXE
PCMATA.SYS
PCMCS.EXE
PCMFDD.EXE
PCMVCD.386
PCMMTD.EXE
PCMSCD.EXE
PENDEV.SYS
POWER.EXE
RAMBOOST.EXE
UMBMONO.SYS
UMBCGA.SYS
UMBHERC.SYS
UMBEMS.SYS

Multiple Configuration Commands

Following are the commands you can use to modify your CONFIG.SYS file so you can select from several different system configurations when you start your computer.

Menu Commands	**Blocks**
INCLUDE	[Common]
MENUCOLOR	[Menu]
MENUDEFAULT	[*User Defined*]
MENUITEM	
SUBMENU	

Conventions Used in this Book

This reference uses particular document and keyboard conventions to help you locate and identify information.

For more information about conventions used with DOS, see "Syntax Conventions" on page 10 in this book.

Document Conventions: To help you locate and identify information easily, this reference uses visual cues and standard text formats. The following typographical conventions are used in this reference:

Type Style	**Used for**
bold	Switches, and any text you must type exactly as it appears. To carry out a command, type the command name and then press **Enter**.
italics	Variables. You can supply the text for any item shown in italics. For example, if you want to use a parameter that calls for a *filename*, you must type the name of the specific file. New terms appear in italics type. The term is explained when it first occurs. Italics type is also used for emphasis in certain examples.
ALL CAPITALS	Commands, directory names, file names, and acronyms. When you type commands, directory names, and file names, you can use lowercase letters.
Initial Capitals	Command programs or the process associated with a command.

Keyboard Convention: The key combination appears in the following format:

Notation	**Meaning**
KEY1+KEY2	A plus sign (+) between key names means you must press the keys at the same time. For example, "Press **Ctrl+C**" means that you press **Ctrl** and hold it down while you press **C**.

Part 1: Command Reference

Chapter 1. Command-Line Basics

DOS indicates the command line by displaying the command prompt, for example:

```
C:\>
```

You type commands at the prompt to specify tasks you want DOS to perform. For example, if you want the DOS version number displayed, do the following:

1. Type `ver`
2. Press **Enter**

   The software version number appears.

Each command contains a set of instructions. For example, when you use the VER command, you instruct DOS to display information about the DOS version number. A command can be a word (TIME) or an abbreviation (DIR). To carry out a command, you type the command and then press **Enter**.

Parts of a Command

A DOS command has up to three parts. Every command has a *command name.* Some commands require one or more *parameters* that identify the object you want DOS to act on. Some commands also include one or more *switches,* that modify the action being performed.

The Command Name

The command name, which you type first, states the action you want DOS to carry out. Some commands (such as the CLS command, that clears your screen) consist only of a command name. Most DOS commands, however, require more than a name.

Parameters

DOS sometimes requires additional information, that you must specify in one or more parameters after the command name. A parameter defines the object you want DOS to act on. For example, the DEL (ERASE) command requires a parameter that names the file you want to delete. Suppose, for example, you want to delete a file named NOTES.TXT. Here is what you would type:

```
del notes.txt
```

Some commands require more than one parameter. For example, to rename a file by using the RENAME (REN) command, you must include the original name of the file in addition to the new name. The following command changes LETTER.TXT to MEMO.TXT:

```
ren letter.txt memo.txt
```

With some commands, parameters are optional. For example, you can use the DIR command without a parameter to list files in the directory you are currently using. Or you can include a parameter (a different drive, for example) to list files in a different directory.

Switches

A switch is a forward slash (/) usually followed by a single letter or number. Switches are used to modify the way a command performs a task. For example, suppose you want to use the DIR command to view a listing of a directory that contains a large number of files. When you type the DIR command by itself, the filenames scroll by so rapidly on the screen that you cannot read them all. If you add the **/p** switch, you can view the list of files one screen at a time.

Some DOS commands do not have any switches, whereas others have several. If a command has more than one switch, you type them one after the other. You can separate switches with a space, but the space is optional.

Typing a Command

The flashing underscore on the command line is the *cursor.* The cursor shows you where to type the command. When you type a character, the cursor moves one space to the right. If you make a mistake, press Backspace to delete a character to the left of the cursor. You can type a command in uppercase or lowercase letters. Unless otherwise specified, you must use a space to separate a command from its parameters.

If you want to retype a command, press **Esc**. The cursor moves to the beginning of the next line, and you can start over. Anything that you typed before you pressed **Esc** is ignored.

Shortcuts to Typing a Command

DOS has editing keys that change or repeat a command that you have typed. Two of the most commonly used keys are **F1** and **F3**:

F1 Displays the previous command one character at a time

F3 Displays all of the previous command

Suppose you type the following:

```
DIR a: #
```

Because there is an extra character at the end of this command, DOS displays an error message. If you press **F3**, the command reappears. Press **Backspace** to erase the number sign (#), and then press **Enter** to see the directory listing.

Suppose you want to retype the same command, substituting drive B for drive A. By pressing **F1** four times, the first part of the command, DIR, appears. You then type **b**, and press **F1** once. The colon (:) appears. If your system has a B drive, press **Enter** to see a directory listing. Otherwise, press **Esc**

DOS has a DOSkey program that retrieves, modifies, and reuses commands.

To install DOSkey, type `doskey` at the command prompt. If DOSkey has not already been installed, the following message appears when you type doskey:

```
DOSKey installed.
```

You can now retrieve and edit commands you type. For example, suppose you type the following three commands:

```
type tuesday
date
time
```

The first command displays the contents of a file named TUESDAY; the second displays the current date; the third displays the current time. All of these commands are stored in the temporary memory of your system.

You can use several methods to retrieve these commands when DOSkey is installed. The easiest method is to press the **Up Arrow** key. If you press the **Up Arrow** key once, the most recent command (TIME) is displayed at the command prompt. Press the **Up Arrow** key two more times to view the first command:

```
type tuesday
```

You can press **Enter** if you want the command to be carried out again, or you can edit the command. For example, press the **Home** key to move the cursor back to the beginning of the line, and type the REN command over the TYPE command. Press the **Del** key to erase the **e** left from the TYPE command. Then, press the **End** key to move the cursor to the end of the line, press the **Spacebar**, and type monday as your new name for the file. The edited command looks like this:

```
ren tuesday monday
```

To carry out the revised command, press **Enter**.

How DOS Responds to a Command

DOS responds to commands in various ways. DOS might display a message indicating that the command has been successfully completed or that you did not type the command correctly.

When you type some commands, DOS prompts you for more information. For example, if you type the TIME command, DOS displays the following prompt:

```
Current time is: 9:52:18:34a
Enter new time:
```

In response, specify the new time or press **Enter**.

Sometimes DOS prompts you to verify a command. For example, suppose you use the following DEL command with wildcards to delete all files in the C:\TMP directory:

```
del c:\tmp\*.*
```

DOS displays the following message:

```
All files in directory will be deleted!
Are you sure (Y/N)?
```

If you do not want to delete all the files, type **n**. If you do want to delete all files, type **y**.

Sometimes DOS displays the results of a command. For example, when you use the COPY command to instruct DOS to copy a particular file, DOS displays the following information:

```
   1 file(s) copied
```

Sometimes you receive an error message indicating that DOS does not recognize the command you typed. If you misspelled the command, type it again and press **Enter**. If the command exists and you typed it correctly, you might have to change directories or specify the directory where the program file is located.

Stopping or Canceling a Command

You can temporarily stop the output of a command by pressing the **Ctrl+S** or the **Pause** keys. Press any key except **Pause** to restart the output of the command. You can stop and restart the output of a command as many times as you want.

If you want to stop DOS from completing a command, press **Ctrl+Break** or **Ctrl+C**. The command is canceled, and the command prompt is displayed.

Note: Any action DOS carries out before you press **Ctrl+Break** or **Ctrl+C** cannot be undone.

Designating a Disk Drive

The *current drive* appears as the first letter of the command prompt. On most systems, if the letter is A or B, one of the diskette drives is the current drive. If the letter is C, the hard disk drive is current. Some systems have additional drives as well.

If the files or directories you want to work with are on a disk in the current drive, you do not need to specify the drive. If the files or directories are not located on the current drive, you can either specify the drive as part of a command or change the current drive.

To change the current drive, type the letter of the drive followed by a colon. For example, to change the current drive from C to A, type the following at the command prompt:

```
   a:
```

To specify another drive, include the drive letter with the command. For example, suppose the current drive is C. To view a list of files on a diskette in drive A, type **a** followed by a colon as a parameter in the DIR command:

```
   dir a:
```

Internal and External Commands

DOS loads some commands into memory when you start your system. These *internal* commands are in a file called COMMAND.COM. Some internal commands are DIR, DEL, DATE, and TIME.

DOS stores *external* commands in files on a disk and transfers them from disk to memory as you use them. Two external commands are CHKDSK and FORMAT. If you set up DOS on a system with a hard disk and you use the default directory, the external-command files are placed in the \DOS directory. If you move the files, indicate their new location by using the PATH command. For information about the PATH command, see Chapter 2, "Commands" on page 9 in this book.

To determine whether a specific command is internal or external, see "Types of Commands" on page 9.

Getting Help with a Command

Online help is available for all DOS commands. The help describes the purpose of the command you specify and provides a summary of its parameters and switches. To get online help for a command, type the command name followed by the **/?** switch, or type **help** followed by the command name.

For example, to get help for the DEL command, you could type this:

```
del /?
```

Or you could type the following, with the same result:

```
help del
```

DOS displays the following help for the DEL command:

```
 Deletes one or more files.

DEL [drive:][path]filename [/P]
ERASE [drive:][path]filename [/P]

 [drive:][path]filename
                Specifies the file(s) to delete. Specify multiple
                files by using wildcards.
   /P           Prompts for confirmation before deleting each file.
```

If you type `help` without a command name, DOS displays a list of all DOS commands and their purpose.

Chapter 2. Commands

This is a complete reference for the commands supplied with DOS. Included are descriptions of the types of commands, an explanation of command syntax, and a description of each command.

Command Types

The following command types can be used with DOS:

- DOS
- Batch
- CONFIG.SYS
- Internal
- External
- Network
- Windows

In this book, the command type follows the description of the command. For example, the command type for COPY appears as follows:

```
COPY

Copies one or more files to another location.

This command can also be used to combine files.
When more than one file is copied, DOS displays
each file name as the file is...

Type

DOS
```

More than one type can apply to a given command. A description of each command type follows:

DOS
DOS commands are basic instructions provided with DOS. For an introduction to using DOS commands, see Chapter 1, "Command-Line Basics" on page 3 in this book.

Batch
Batch commands are *internal commands* that you can use to direct how a batch program runs. For a description of internal commands, see the "Internal" topic on page 9.

CONFIG.SYS
CONFIG.SYS commands are commands that you can use to customize your system. These commands are useful for such tasks as installing device drivers, setting limits on files and buffers, and carrying out DOS commands during CONFIG.SYS processing.

Internal
Internal commands are stored in the COMMAND.COM file, which is loaded into memory when you start your system. They include the simpler, more commonly used commands you need on a regular basis. Because internal commands are part of COMMAND.COM, you never see

their names in a directory listing. These commands remain resident in memory and are available to you at all times.

External

External commands exist as separate files on your disk. When you use the DIR command to view the files on your DOS system disk, you see the external commands in the list of file and directory names. The file name of an external command has a .COM, .EXE, or .BAT extension.

External commands sometimes perform in ways similar to programs. For that reason, some users may refer to them as utilities. However, in this book, they are called commands.

Network

Not all DOS commands are designed to be used on a network. If a command can be used with a network, it is indicated under the heading "Type" for that command.

Windows

DOS contains several commands used with Windows (Version 3.1 only). If you have Windows installed on your computer, you can use these commands from the DOS command-line prompt or from the IBM Tools Group on the Windows desktop.

Syntax Conventions

Syntax represents the order in which you must type a DOS command and any parameters and switches that follow it. Elements that appear in bold type must be typed exactly as they appear in the syntax line; elements that appear in italics are placeholders representing specific information that you supply.

Unless specified otherwise, you can type commands, parameters, and switches in either uppercase or lowercase letters.

The following is a sample syntax line:

```
   1   2  3        4    5        6     7       8
   ↓   ↓  ↓        ↓    ↓        ↓     ↓       ↓
sample [+r|-r][drive:][path]filename[ ...] [options]
```

The meaning of these elements is as follows:

Number	Element	Meaning
1	**sample**	Specifies the name of the command.
2	[ ]	Indicates an item that is optional. To include the optional information described within the brackets, type only the information, not the brackets themselves.

<table>
<tr><th>Number</th><th>Element</th><th>Meaning</th></tr>
<tr><td>3</td><td>|</td><td>Separates two mutually exclusive choices in a syntax line, as shown in the following example:
<b>break [on|off]</b>
Type only one of these choices (that is, <b>break on</b> or <b>break off);</b> do not type the pipe (|) itself.
<b>Note:</b> The pipe (|) is used as a redirection symbol. To the computer, the two pipes (| and |) are equivalent—that is, you press the same key to type any pipe. Before using a pipe for redirection, you should set the TEMP environment variable in your AUTOEXEC.BAT file.</td></tr>
<tr><td>4</td><td><i>drive</i>:</td><td>Specifies the name of a hard disk drive or diskette drive. To carry out an external command when its file is not on the disk in the current drive or in the search path, you must specify the correct drive. You never need to specify the drive of an internal command.</td></tr>
<tr><td>5</td><td><i>path</i></td><td>Specifies the route the operating system is to follow through the directory structure to locate a directory or file. You need to specify a path with a filename only if the file is not in the current directory.</td></tr>
<tr><td>6</td><td><i>filename</i></td><td>Specifies the name of a file. A filename can be up to eight characters long and can be followed by a period (.) and an extension of up to three characters (for example, YOURFILE.TXT). In this book, lowercase filenames are used in example blocks and uppercase filenames are used in other sections. You can type either uppercase or lowercase filenames. You cannot specify a device name or drive letter for <i>filename</i>.</td></tr>
<tr><td>7</td><td>...</td><td>Indicates that the previous parameter or switch can be repeated several times in a command. Type only the information, not the ellipsis (...) itself.</td></tr>
<tr><td>8</td><td><i>options</i></td><td>Specifies one or more optional command parameters or switches. A switch usually begins with a slash—for example, <b>/p</b>.</td></tr>
</table>

Other placeholders used in syntax lines in this manual include the following:

Placeholder	Meaning
source	Specifies the location of data to be transferred to a specified destination or used as input to a command. *Source* can consist of a drive letter and colon, a directory name, a filename, or a combination.
destination	Specifies a location to which the data specified by *source* is to be transferred. *Destination* can consist of a drive letter and colon, a directory name, a filename, or a combination.
string	Specifies a group of characters to be treated as a unit. A string can include letters, numbers, spaces, or any other characters and is usually enclosed in double quotation marks. Some commands, such as FIND, work with strings of text.

Online Help for DOS Commands

DOS includes online help for DOS commands. To get help with the syntax, parameters, and switches of any DOS command, type the command name followed by **/?** on the command line or type `help` followed by the command name. For example, for help information about the COPY command, type one of the following:

```
copy /?
```

```
help copy
```

DOS displays information about the command syntax, parameters, and switches.

To see a list of all DOS commands and a brief description of the purpose of each command, type `help` with no parameters or switches.

Chapter 3. Command Type Quick Reference Tables

The following tables are provided to give a quick overview of the Batch, CONFIG.SYS and Device Driver commands that DOS uses, the primary task or function that is performed by the command, and a page reference where more detail can be found.

For an alphabetical listing and details of all DOS commands, see Chapter 4, "DOS Commands" on page 19.

Batch Commands

Any DOS command you use at the DOS command prompt or in DOS Shell can also be put in a batch program. In addition, there are DOS commands that are specially designed for batch programs. The commands and their functions are as follows:

Command	Action	See Page #
CALL	Runs a second batch program and then returns to the first one.	30
CHOICE	Prompts you to choose from a set of choices, waits until you make a choice by pressing a key, and beeps if you select a key that is not among the available choices.	37
ECHO	Displays messages on your screen or turns the ECHO feature on or off.	94
FOR	Carries out a command for a group of files or directories.	113
GOTO	Switches to commands in another part of your batch program and continues processing commands from that point.	119
IF	Carries out a command based on the result of a condition.	126
PAUSE	Temporarily stops your batch program from running; your program starts running again when you press any key.	176
REM	Annotates your batch program so that you can remember what each part of the program does.	197
SHIFT	Changes the position of replaceable parameters.	215
@	Is placed in front of a command in your batch program and prevents the single command from being displayed.	19

For more information about using Batch commands and working with batches, see the *PC DOS User's Guide.*

CONFIG.SYS Commands

The CONFIG.SYS file is a text file that contains special commands. These commands set up your computer's hardware components (such as memory, keyboard, mouse, and printer) so that DOS and applications can use them. When DOS starts, it processes the commands in the CONFIG.SYS file.

CONFIG.SYS File Editing

To edit the CONFIG.SYS file, use a text editor, such as the DOS E editor, that can save files as unformatted (ASCII) text. Do not edit the CONFIG.SYS file using a word processor that saves files in a special document format. If you do, your computer might not start.

Because the CONFIG.SYS file controls how DOS starts, DOS reads it only when you start your computer. Therefore, after changing the CONFIG.SYS file, you must restart your computer for your changes to take affect.

To make changes to your CONFIG.SYS file:

1. Make a copy of your CONFIG.SYS file on a separate diskette before you make any changes.
2. Edit the CONFIG.SYS file using the DOS E editor.
3. Add or change CONFIG.SYS commands as necessary. Each CONFIG.SYS command must begin on a separate line.
4. When you have finished editing the CONFIG.SYS file, save your changes and exit from the text editor.
5. Restart your system by pressing CTRL+ALT+DEL so that your changes can take affect.

CONFIG.SYS Command Table

The following table lists the commands that can be inserted in your CONFIG.SYS file. Some of these commands are used only when you are using multiple configurations, some are special character commands, and others can be used in your AUTOEXEC.BAT file or typed from the command line prompt. A typical CONFIG.SYS file contains some, but not all, of these commands.

Command	Action	See Page #
BREAK	Specifies whether DOS is to check periodically for the CTRL+C or CTRL+BREAK key combination.	28
BUFFERS=	Specifies how much memory DOS reserves for transferring information to and from hard disks or diskettes.	29
COUNTRY=	Sets the language conventions for your system.	51
DEVICE=	Loads an installable device driver—a program that controls a hardware component, such as a mouse or memory board.	69
DEVICEHIGH=	Loads an installable device driver into the upper memory area.	70

Command	Action	See Page #
DOS=	Specifies whether DOS will use the high memory area (HMA) and whether to provide access to the upper memory area.	82
DRIVPARM=	Sets the characteristics of a disk drive.	90
FCBS=	Specifies the number of file control blocks (FCBs) that DOS can have open at the same time. A *file control block* is a data structure that stores information about a file.	108
FILES=	Specifies how many files can be open at a time.	110
INCLUDE=	Instructs DOS to carry out the commands in another configuration block as well as the commands in the current block. Used only with multiple configurations.	128
INSTALL=	Loads a memory-resident program (also called a terminate-and-stay-resident program, or TSR).	128
INSTALLHIGH=	Loads a memory-resident program into Upper Memory Blocks (UMBs). If upper memory is not available, this command functions like the INSTALL command.	129
LASTDRIVE=	Sets the number of valid drive letters.	145
MENUCOLOR=	Sets the text and background colors for the menu. This command can only be used with multiple configurations.	152
MENUDEFAULT=	Specifies the default menu item. This command is optional. If the [Menu] block does not contain a MENUDEFAULT command, the default is set to item 1. This command can only be used with multiple configurations.	154
MENUITEM=	Defines a menu item. The command specifies the configuration block associated with that item and, optionally, the menu text for that item. This command can only be used with multiple configurations.	155
NUMLOCK=	Specifies whether the Num Lock setting of the numeric keypad is initially set to ON or OFF.	174
REM	Indicates that the current line is a descriptive comment. DOS does not carry out REM commands.	197
SET	Sets the value of an environment variable, such as PROMPT or TEMP.	206
SHELL=	Specifies the command interpreter (a program that displays the command prompt where you type commands) that you want DOS to use.	214
STACKS=	Specifies how much memory to reserve for processing hardware interrupts.	222
SUBMENU=	Specifies a menu item that displays another set of choices. The command specifies another menu block that defines the choices of the submenu. This command can only be used with multiple configurations.	223
SWITCHES=	Specifies special options using these switches: **/K**, **/W**, **/N**, and **/F**.	225

Command	Action	See Page #
;	Specifies that the current line is a descriptive comment and should not be carried out. Insert this character at the beginning of the line.	19
?	Specifies that DOS is to ask for confirmation before carrying out the current command. Insert this character just before the equal sign (=). For example, to have DOS ask for confirmation before carrying out the DOS=HIGH command, you would change the command to read DOS?=HIGH. This command can be used only with certain CONFIG.SYS commands.	20

For more information about working with multiple configurations and editing your CONFIG.SYS file, see *PC DOS User's Guide*.

Device Driver Commands

Each hardware component of your computer is called a *device*. Your computer's keyboard, mouse, display, printer, disk drives, and memory boards are all devices. Each device has its own characteristics that can be customized.

DOS uses a special program called a *device driver* to control each device. For example, DOS uses a built-in device driver to control how information is read to and from a diskette drive. DOS has built-in device drivers for your keyboard, display, hard and diskette drives, and communication ports. Because these device drivers are built in, you do not have to do anything special to use them. You can customize certain features of these devices by using CONFIG.SYS commands.

Other devices, such as memory boards or a mouse, have device drivers that are not built into DOS. Such a device driver is called an *installable* device driver because you install it by adding a command to your CONFIG.SYS file. Many hardware devices come with their own device drivers.

See Chapter 5, "Working With Device Drivers" on page 245 for a more detailed discussion of device drivers.

DOS comes with the following installable device drivers.

Command	Action	See Page #
ANSI.SYS	Supports American National Standards Institute (ANSI) terminal emulation.	247
CMOSCLK.SYS	Replaces the default DOS clock so that any request for the current date and time accesses CMOSCLK.SYS instead of the DOS system clock. Normally used only if your current DOS system clock is not keeping the correct date.	253
DISPLAY.SYS	Supports code page switching for displays.	254
DRIVER.SYS	Creates a logical drive that you can use to refer to a physical disk driver and specifies parameters for a drive not supported by your computer's ROM BIOS.	255

Command	Action	See Page #
EGA.SYS	Saves and restores the display when a graphics program is used with an EGA monitor.	257
EMM386.EXE	Simulates expanded memory and provides access to the upper memory area on a computer with at least an 80386 processor and extended memory.	258
HIMEM.SYS	Manages the use of extended memory on a computer with at least an 80286 processor and extended memory. DOS Setup installs this device driver automatically on such systems.	263
INTERLNK.EXE	Redirects requests for operations on one or more Interlnk client drives or printer ports to one or more drives or printer ports on the Interlnk server. You must install the INTERLNK.EXE device driver before you can use the INTERLNK and INTERSVR commands.	267
PCMATA.SYS	Provides virtual block device driver support for PCMCIA ATA fixed disk cards if they are formatted with a FAT-compatible disk structure.	268
PCMCS.EXE	Loads PCMCIA Card Services support. Can be loaded either as a terminate-and-stay-resident program from the DOS command prompt or as a device driver when included in the CONFIG.SYS file.	270
PCMFDD.EXE	Emulates diskette drives from PCMCIA cards. Can be loaded either as a terminate-and-stay-resident program from the DOS command prompt or as a device driver when included in the CONFIG.SYS file.	271
PCMMTD.EXE	Installs one or more Memory Technology Drivers (MTDs) so that PCMCIA cards with different memory technologies can be loaded either as a terminate-and-stay-resident program from the DOS command prompt or as a device driver when included in the CONFIG.SYS file.	272
PCMSCD.EXE	Installs one or more Super Client Drivers (SCDs) so that PCMCIA cards with different devices, such as fax machines or modems, can be supported. Can be loaded either as a terminate-and-stay-resident program from the DOS command prompt or as a device driver when included in the CONFIG.SYS file.	273
PCMVCD.386	PCMVCD.386 is the Windows VxD (virtual device driver) for PC Card support in 386-enhanced mode. It is a replacement module for the Windows VCD (virtual COMM driver). PCMVCD.386 allows fax and modem cards inserted into a PCMCIA socket to be available to all sessions under Windows. During installation of DOS, if you select PCMCIA support, PCMVCD.386 gets copied in your DOS directory. However, if you have Windows and want Phoenix PCMCIA Support, you must manually edit the SYSTEM.INI file before you can have Phoenix PCMCIA Support under Windows.	274
PENDEV.SYS	Provides the PenDOS application programming interface (API) for PenDOS applications.	276

Command	Action	See Page #
POWER.EXE	Provides the ability to reduce the consumption of power when your applications and devices are idle.	276
PRINTER.SYS	Supports code page switching for printers.	277
RAMDRIVE.SYS	Simulates a hard disk drive by creating a virtual disk drive in your system's random access memory (RAM).	280
RAMBOOST.EXE	Increases your computer's conventional memory and makes using EMM386.EXE less complex. Automatically analyzes your computer's current configuration and then reconfigures it automatically to load terminate-and-stay-resident programs and device drivers into upper memory blocks. This eliminates the need to manually edit your CONFIG.SYS or AUTOEXEC.BAT files because it maintains a data file called RAMBOOST.INI to track the programs and device drivers that need to be loaded into upper memory.	278
SMARTDRV.EXE	Loads the SMARTDRV.EXE device driver to perform double buffering.	282
SETVER.EXE	Loads the DOS version table into memory.	282
UMBCGA.SYS	Maps the video memory of a color adapter as Upper Memory Blocks (UMBs) that are used for loading programs with the LOADHIGH command if a CGA, EGA, or VGA adapter is also present. It provides approximately 16K of extra memory.	284
UMBEMS.SYS	Maps a 64K block of EMS memory as Upper Memory Blocks (UMBs) that are used for loading programs with the LOADHIGH command if an EMS card with at least 64K of EMS is also present in a machine. It provides approximately 64K of extra memory. Works on any class of machine with any level of EMS memory driver, but the EMS 3.2 level card is preferred because it uses less memory when loaded.	286
UMBHERC.SYS	Maps the video memory of a Hercules adapter as Upper Memory Blocks (UMBs) that are used for loading programs with the LOADHIGH command if a Hercules graphics adapter or a Hercules Graphics Adapter Plus is also present. It provides approximately 60K of extra memory.	285
UMBMONO.SYS	Maps the video memory of the monochrome adapter as Upper Memory Blocks (UMBs) that are used for loading programs with the LOADHIGH command if a CGA, EGA, or VGA adapter is also present. It provides approximately 4K of extra memory.	283

Chapter 4. DOS Commands

The commands found in this section are those that can be typed from the command prompt on your system or inserted into your CONFIG.SYS or AUTOEXEC.BAT files.

@

The ATSIGN (@) command is used in batch programs. It is placed in front of a command in your batch program and prevents the single command from being displayed.

Type

DOS, Batch

Syntax

@ [*batch line*]

Parameters

[*batch line*]
Any executable line in your batch program.

Related Information

For additional information about working with batch programs, see the *PC DOS User's Guide.*

;

The semi-colon (;) command is used in your CONFIG.SYS file only. It specifies that the current line is a descriptive comment and should not be carried out. Insert this character at the beginning of the line.

Type

CONFIG.SYS

Syntax

; [*comment line*]

Parameters

[comment line]
Any information line in the CONFIG.SYS file that you do not want processed.

?

The question mark (?) can be used with certain CONFIG.SYS commands. It specifies that DOS is to pause and ask for confirmation before processing the current command.

Type

CONFIG.SYS

Syntax

[*command*]**?=**

Parameters

command
A CONFIG.SYS file command. See the following list.

Notes

CONFIG.SYS commands that can use the ? command

The following CONFIG.SYS commands can use the ? command:

BREAK=　　BUFFERS=

DEVICE=　　DEVICEHIGH=

DOS=　　FCBS=

FILES=　　INSTALL=

INSTALLHIGH=　　LASTDRIVE=

STACKS=　　SWITCHES=

Where to place the ? command

Insert this character just before the equal sign (=).

Examples

For example, to have DOS ask for confirmation before carrying out the DOS=HIGH command, you would change the command to read DOS?=HIGH.

ANSI.SYS

This is an installable device driver that supports American National Standards Institute (ANSI) terminal emulation.

Related Information

For detail information see command “ANSI.SYS” on page 247.

APPEND

Enables programs to open data files in specified directories as if these files were in the current directory.

The specified directories are called *appended directories* because, for the sake of opening data files, they can be found as if they were appended to the current directory.

Type

DOS, External, Network

Syntax

append [[*drive*:]*path*[;...]][**/x**[**:on**|**:off**]][**/path:on**|**/path:off**] [**/e**]

To display the list of appended directories, type the following:

```
append
```

To clear the appended directory list, type the following:

```
append;
```

Parameters

[*drive*:]*path*
Specifies the drive (if other than the current drive) and directory that you want to append to the current directory. You can specify multiple entries of [*drive*:]*path*, separating the entries with semicolons.

; When used by itself (APPEND ;), clears the existing list of appended directories.

Switches

/x[:on | :off]
Specifies whether DOS is to search appended directories when executing programs. If you use the **/x:on** switch, the program searches appended directories. If you use the **/x:off** switch (the default setting), the program does not search appended directories. You can

abbreviate **/x:on** to **/x**. If you want to specify **x:on**, you must do it the first time you use APPEND after starting your system. After that, you can switch between **x:on** and **x:off**.

/path:on|/path:off
Specifies whether a program is to search appended directories for a data file when a path is already included with the name of the file the program is looking for. The default setting is **/path:on**.

/e Assigns the list of appended directories to an environment variable named *append*. This switch can be used only the first time you use APPEND after starting your system. If you use **/e**, you can use the SET command to display the list of appended directories. For information about environment variables, see the "SET" on page 206.

Notes

Storing the list of appended directories in the environment

You can use the **/e** switch with APPEND to assign the list of appended directories to an environment variable named *append*. To do this, first use the APPEND command with only the **/e** switch. Then use APPEND again, this time including the directories you want to append. You cannot specify **/e** and [*drive*:]*path* on the same command line.

Specifying multiple appended directories

To append more than one directory, separate multiple entries with semicolons. If you use the APPEND command with the [*drive*:]*path* parameters again, the specified directory or directories replace any directories specified in a previous APPEND command.

Appended directories and the DIR command

The DIR command does not add filenames from appended directories to directory listings produced by the DIR command.

File name conflicts

If a file in an appended directory has the same name as a file in the current directory, programs open the file in the current directory.

Using append with programs that create new files

When a program opens a file in an appended directory, the file can be found as if it were in the current directory. If the program then saves the file by creating a new file with the same name, the new file is created in the current directory (not the appended directory). APPEND is appropriately used for data files that are not to be modified or that are to be modified without creating new copies of the files. Database programs often modify data files without making new copies. Text editors and word processors, however, usually save modified data files by making new copies. To avoid confusion, you might want to avoid using APPEND with these programs.

Using the /x:on switch and the PATH command

When **/x:on** is specified, you can run a program located in an appended directory by typing the program name at the command prompt. Usually, you use the PATH command to specify directories that contain programs. However, when your program is in an appended directory, you do not need to use the PATH command to specify that directory. DOS finds a program in an appended directory by following the usual order in which DOS searches for the program first in the current directory, then in the appended directories, and then in the search path.

DOS functions that always use appended directories

Even when the **/x:on** switch is not specified, appended directories are used when programs call the following DOS Interrupt 21H functions:

- Open File (0FH)
- Open File Handle (3DH)
- Get File Size (23H)

When **/x:on** is specified, appended directories are used when programs call any of the Interrupt 21H functions in the preceding list or any of the Interrupt 21H functions in the following list:

- Find First Entry (11H)
- Find First File (4EH)
- Execute Program (EXEC) (4BH)

Using APPEND with the ASSIGN command

If you use both the APPEND and ASSIGN commands, you must use APPEND first, even if the commands affect different drives. When you use the ASSIGN command to assign a different drive letter to an existing disk drive, you cannot use the APPEND command to append directories that use the new drive letter.

Using APPEND with network drives

You can use the APPEND command to append directories that are located on network drives.

Examples

To allow programs to open data files in a directory named LETTERS on the diskette in drive B and in a directory named REPORTS on the diskette in drive A as if the files were in the current directory, type the following command:

```
append b:\letters;a:\reports
```

To append the same directories and keep a copy of the list of appended directories in the DOS environment, type the following commands:

```
append /e
append b:\letters;a:\reports
```

These must be the first APPEND commands you use after starting your system.

Related Information

For information about setting a search path for executable files, see command "PATH" on page 175.

ASSIGN

Redirects requests for disk operations on one drive to a different drive.

Some older programs can read and write files only on drives A and B. With the ASSIGN command, you can redirect disk operations for those programs so that you can read and write files on drives other than A and B.

Type

DOS, External, Network

Syntax

assign [*x*[:]=*y*[:][...]]

assign /status

Type `assign` without parameters to reset all drive letters to their original assignments.

Parameters

x Specifies the drive from which you want to redirect read and write operations. This value must be a letter. The use of the colon (:) is optional.

y Specifies the existing drive to which you want to redirect read and write operations. This value must be a letter. The use of the colon (:) is optional.

Switches

/status
Lists current assignments. You can abbreviate this switch as **/sta** or **/s**.

Notes

Invalid uses of ASSIGN

You must not assign the drive letter of your hard disk to another drive. You should not use ASSIGN for a drive that is in use by a program.

You cannot use the drive letter of a hard disk drive that does not exist for either the *x* or the *y* parameter.

Avoid the use of ASSIGN in the following cases:

- With commands requiring drive information (JOIN, LABEL, RESTORE, SUBST)
- With the DISKCOPY and FORMAT commands, which ignore drive reassignments
- During typical use of DOS, unless a program cannot read and write files on the specified drive

Using ASSIGN with the APPEND command

If you use both the ASSIGN and APPEND commands, you must use APPEND first, even if the commands affect different drives.

Using ASSIGN for network drives

You can use the ASSIGN command for network drives.

Canceling a previous assignment as the result of a new assignment

Assigning a drive letter to a drive cancels previous assignments to it. For example, you assign drive A to drive letter C as follows:

```
assign a=c
```

Later you assign drive B to drive letter C, as the following example shows:

```
assign b=c
```

As a result, drive A is no longer assigned to drive letter C.

Using the SUBST command instead of ASSIGN

You can use the SUBST command instead of ASSIGN. The following commands produce the same result:

```
assign a=c
subst a: c:\
```

Examples

Suppose you want to use drive C to read and write files, but your program requires you to put your program diskette into drive A and your data diskette into drive B. To reassign the drive letters A and B to drive C, type the following command:

```
assign a=c b=c
```

This command causes DOS to look for your program and data files on drive C.

To reset all drive letters to their original drives, type the ASSIGN command without parameters at the command prompt.

Related Information

For information about associating a drive letter with a path, see command “SUBST” on page 224.

ATTRIB

Displays or changes file attributes.

This command displays, sets, or removes the read-only, archive, system, and hidden attributes assigned to files.

Type

DOS, External, Network

Syntax

attrib [**+r**|**–r**] [**+a**|**–a**] [**+s**|**–s**] [**+h**|**–h**] [[*drive*:][*path*]*filename*] [**/s**]

To display all attributes of all files in the current directory, use the following syntax:

attrib

Parameters

[*drive*:][*path*]*filename*
Specifies the location and name of the file or set of files you want to process.

Switches

+r Sets the read-only file attribute.

–r Clears the read-only file attribute.

+a Sets the archive file attribute.

–a Clears the archive file attribute.

+s Sets the file as a system file.

–s Clears the system file attribute.

+h Sets the file as a hidden file.

–h Clears the hidden file attribute.

/s Processes files in the current directory and all of its subdirectories.

Notes

Groups of files

You can use wildcards (? and *) with the *filename* parameter to display or change the attributes for a group of files. If a file has the system or hidden attribute set, you must clear that attribute before you can change any other attributes for that file.

Archive attributes

The archive attribute (**+a and -a**) is used to mark files that have changed since they were previously backed up. The RESTORE and XCOPY commands use these archive attributes. For information about archive attributes, see commands "RESTORE" on page 201 and "XCOPY" on page 240.

Examples

To display the attributes of a file named NEWS86 located on the current drive, type the following command:

```
attrib news86
```

To assign the read-only attribute to the file REPORT.TXT, type the following command:

```
attrib +r report.txt
```

To remove the read-only attribute from files in the \USER\JONES directory on a disk in drive B and from files in any subdirectories of \USER\JONES, type the following command:

```
attrib -r b:\user\jones\*.* /s
```

As a final example, suppose you want to give an associate a disk containing all files in the default directory on a diskette in drive A except files with the .BAK extension. Because you can use XCOPY to copy only those files marked with the archive attribute, you need to set the archive attribute for those files you want to copy. To do this, you use the following two commands to set the archive attribute for all files on drive A and then to clear the attribute for those files with the .BAK extension:

```
attrib +a a:*.*

attrib -a a:*.bak
```

Next, use the XCOPY command to copy the files from the diskette in drive A to the diskette in drive B. The **/a** switch in the following command causes XCOPY to copy only those files marked with the archive attribute:

```
xcopy a: b: /a
```

Related Information

For more information about copying files and directories, see command "XCOPY" on page 240.

BREAK

Sets or clears extended **Ctrl+C** checking.

You can press **Ctrl+C** to stop a program or an activity (file sorting, for example). Typically, DOS checks for Ctrl+C only while it reads from the keyboard or writes to the screen or a printer. If you set BREAK to On, you extend Ctrl+C checking to other functions, such as disk read and write operations.

Type

DOS, CONFIG.SYS

Syntax

break [**on**|**off**]

To display the current BREAK setting, type the following at the command prompt:

```
break
```

Parameters

on|**off** Turns extended Ctrl+C checking on or off.

Notes

Including BREAK in CONFIG.SYS

The default setting for BREAK is Off. You can include the BREAK command in your CONFIG.SYS file to enable extended Ctrl+C checking every time you start your system.

Edit your CONFIG.SYS file and add the following line:

```
break=on
```

The disadvantage of setting **break=on** is that it slightly decreases the speed of your system.

Examples

To specify that DOS is to check for Ctrl+C only while it is reading from the keyboard or writing to the screen or printer, type the following command:

```
break off
```

To specify that DOS is to check for Ctrl+C while it is reading from a disk or the keyboard or writing to a disk or the screen, type the following command:

```
break on
```

BUFFERS

Allocates memory for a specified number of disk buffers when your system starts.

Type

CONFIG.SYS

Syntax

buffers=*n*[,*m*]

Parameters

n Specifies the number of disk buffers. The value of *n* must be in the range 1 through 99.

m Specifies the number of buffers in the secondary buffer cache. The value of *m* must be in the range 1 through 8.

Notes

Default settings

The default setting for the number of disk buffers depends upon the configuration of your system, as shown in the following table:

Configuration	**Buffers (*n*)**	**Bytes**
< 128K of RAM, 360K disk	2	—
< 128K of RAM, > 360K disk	3	—
128K to 255K of RAM	5	2672
256K to 511K of RAM	10	5328
512K to 640K of RAM	15	7984

The default setting for the number of buffers in the secondary cache (*m*) is 1.

If you specify an invalid value for *n* or *m*, BUFFERS uses the default value instead.

How DOS uses buffers

DOS uses the memory reserved for each disk buffer to hold data during read and write operations. To achieve the best performance with programs such as word processors, specify a value between 10 and 20 for *n*. If you plan to create many subdirectories, you might want to increase the number of buffers to 20 or 30. Each buffer requires approximately 532 bytes of memory. Therefore, the more buffers you have, the less memory you have available for programs.

If DOS is in the high memory area (HMA), the buffers are also in HMA. This leaves more conventional memory for your program.

Usage recommendation

When using a disk cache like Smartdrv, a lower number of buffers can be used than can be used on an uncached machine. Some value between 3 - 8 is a good place to start.

Examples

To create 20 disk buffers, include the following command in your CONFIG.SYS file:

```
buffers=20
```

CALL

Calls one batch program from another without causing the parent batch program to stop.

Type

Batch, Internal

Syntax

call [*drive*:][*path*]*filename* [*batch-parameters*]

Parameters

[*drive*:][*path*]*filename*
Specifies the location and name of the batch program you want to call. *Filename* must have a .BAT extension.

batch-parameters
Specifies any command-line information required by the batch program.

Notes

Using *batch-parameters*

Batch-parameters can contain any information that you can pass to a batch program, including switches, file names, the replaceable parameters *%1* through *%9*, and variables such as *%baud%*.

For more information about these variables, see the *PC DOS User's Guide*

Using pipes and redirection symbols

Do not use pipes and redirection symbols with the CALL command.

Making a recursive call

You can create a batch program that calls itself; however, you must provide an exit condition. Otherwise, the parent and child batch programs can loop endlessly.

Examples

To run the CHECKNEW.BAT program from another batch program, include the following command in the parent batch program:

```
call checknew
```

Suppose the parent batch program accepts two replaceable parameters and you want it to pass those parameters to CHECKNEW.BAT. You can use the following command in the parent batch program:

```
call checknew %1 %2
```

CHCP

Displays the number of the active code page, or changes the active code page that DOS is to use for all devices that support code page switching to one of the two prepared system code pages associated with your current country setting.

Type

DOS, Internal, Network

Syntax

chcp [*nnn*]

To display the number of the active code page, use the following syntax:

chcp

Parameters

nnn Specifies the prepared system code pages defined by the COUNTRY command in the CONFIG.SYS file. The following list shows each code page that DOS supports and its country or language:

437	United States
850	Multilingual (Latin I)
852	Multilingual (Latin II)
855	Multilingual (Cyrillic)

857	Turkish
860	Portuguese
861	Icelandic
863	Canadian-French
866	Russian
869	Greek

Notes

Requirements for using the CHCP command

Before you can use the CHCP command, you must specify the location of the COUNTRY.SYS file by using the COUNTRY command in the CONFIG.SYS file and loading the Nlsfunc program into memory.

Assigning a new code page

After you assign a new code page, any program you start uses that new code page. However, any program (not including COMMAND.COM) that you started before assigning the new code page will probably attempt to use the original code page.

Examples

To view the active code-page setting, type the following command:

```
chcp
```

DOS responds with a message similar to the following:

```
Active code page: 437
```

To change the active code page to 850 (Multilingual), type the following command:

```
chcp 850
```

DOS alerts you if the specified code page has not been prepared for your system. The following error message appears:

```
Invalid code page
```

If a device (screen, keyboard, printer) is not prepared for a code page, DOS displays an error message in the following format:

```
Code page 850 not prepared for all devices
```

Related Information

For more information about code pages, see the (set device code pages) commands “COUNTRY” on page 51, “NLSFUNC” on page 173, “DEVICE” on page 69, and “MODE” on page 158.

CHDIR (CD)

Displays the name of the current directory or changes the current directory.

Type

DOS, Internal, Network

Syntax

chdir [*drive*:][*path*]

chdir[..]

cd [*drive*:][*path*]

cd[..]

To display the names of the current drive and directory, use either of the following syntax lines:

chdir

cd

Parameters

[***drive***:][***path***]
Specifies the drive (if other than the current drive) and directory to which you want to change.

.. Specifies that you want to change to the parent directory.

Notes

Changing to the root directory

The *root directory* is the top of the directory hierarchy for a drive. To return to the root directory, type the following command:

```
cd \
```

Using the current directory from a different drive

If you are working in the \USER\JONES directory on drive C and you change to drive D, you can copy files to and from the \USER\JONES directory by specifying only the drive letter C.

Changing the directory on another drive

You can change the current directory on another drive by specifying the drive name on the command line when you use CHDIR.

Examples

Either of the following commands changes your current directory to the directory named PRIMETIM:

```
chdir \primetim
cd \primetim
```

Suppose you have a directory named SPECIALS with a subdirectory named SPONSORS. To change your current directory to \SPECIALS\SPONSORS, type the following command:

```
cd \specials\sponsors
```

Or, if your current directory is \SPECIALS, you can use the following command to change to the \SPECIALS\SPONSORS directory:

```
cd sponsors
```

To change from a subdirectory back to the parent directory, type the following command:

```
cd..
```

To display the name of the current directory, you can use CHDIR or CD without a parameter. For example, if your current directory is \USER\JONES on the disk in drive B, type **chdir** to see the following response:

```
B:\USER\JONES
```

If you are working on drive D and you want to copy all files in the \USER\JONES and \USER\LEWIS directories on drive C to the root directory on drive D, type the following commands:

```
chdir c:\user\jones
copy c:*.* d:\
chdir c:\user\lewis
copy c:*.* d:\
```

If, instead, you want to copy all files in the \USER\JONES and \USER\LEWIS directories to your *current* location on drive D, type the following commands:

```
chdir c:\user\jones
copy c:*.* d:
chdir c:\user\lewis
copy c:*.* d:
```

CHKDSK

Creates and displays a status report for a disk.

The status report shows logical errors found in the file allocation table and file system. If errors exist on the disk, a message is displayed on your screen indicating what is wrong. You should use the CHKDSK command occasionally on each disk to check for errors.

Type

DOS, External

Syntax

chkdsk [*drive*:][[*path*]*filename*] [**/f**] [**/v**]

To display the status of the disk in the current drive, type the following at the command prompt:

```
chkdsk
```

Parameters

***drive*:** Specifies the drive that contains the disk that you want to check.

[*path*]*filename*
Specifies the location and name of a file or set of files that you want to check. You can use wildcards (* and ?) to specify multiple files.

Switches

/f Fixes errors on the disk.

/v Displays the name of each file in every directory as the disk is checked.

Notes

Format of status reports

DOS displays Chkdsk status reports in the following format:

```
Volume Serial Number is B1AF-AFBF

  72214528 bytes total disk space
     73728 bytes in 3 hidden files
     30720 bytes in 12 directories
  11493376 bytes in 386 user files
     61440 bytes in bad sectors
  60555264 bytes available on disk

      2048 bytes in each allocation unit
     35261 total allocation units on disk
     29568 available allocation units on disk

    655360 total bytes memory
    493456 bytes free
```

Fixing disk errors

The CHKDSK command corrects disk errors only if you specify the **/f** switch. Since repairs usually change the file allocation table of a disk and sometimes cause loss of data, you are prompted with a confirmation message similar to the following:

```
10 lost allocation units found in 3 chains.
Convert lost chains to files?
```

If you press the **Y** key, DOS saves each lost chain in the root directory as a file with a name in the format FILE*nnnn*.CHK. When the Chkdsk program finishes, you can check these files to see if they contain data you need. If you press **N**, DOS fixes the disk but does not save the contents of the lost allocation units.

If you do not use the **/f** switch, a message is displayed indicating that a file needs to be fixed.

Using CHKDSK with open files

If you specify the **/f** switch, an error message is displayed if open files are found on the disk. If you do not specify the **/f** switch and open files exist, lost allocation units on the disk might be reported. This could happen if open files have not yet been recorded in the file allocation table. If a large number of allocation units is reported, consider repairing the disk. To avoid problems caused by open files, avoid using the CHKDSK command from another program or from a DOS command interpreter started from Microsoft Windows**.

Using CHKDSK with assigned drives and networks

The CHKDSK command does not work on drives formed by the SUBST, JOIN, or ASSIGN command. You cannot use the CHKDSK command to check a disk on a network drive.

Physical disk errors

The CHKDSK command finds only logical errors in the file system, not physical disk errors. For information about recovering physically damaged files, see the command "RECOVER" on page 196.

Bad disk sectors

When the CHKDSK command is used and bad sectors are found on the disk, they are marked as "bad" when your disk is first prepared for operation. They pose no danger.

Saving a CHKDSK status report to a file

You can save a Chkdsk status report by redirecting the output to a file. Do not use the **/f** switch when you redirect Chkdsk output to a file.

Examples

If you want to check the diskette in drive A and have DOS fix any errors encountered, type the following command:

```
chkdsk a: /f
```

The Chkdsk program pauses and displays messages if it encounters errors. When errors found, you are prompted to specify whether you want DOS to correct the errors. The Chkdsk program finishes by displaying a report showing the status of the disk.

To redirect the output of the Chkdsk program to a file named STATUS, type the following command:

```
chkdsk a: > status
```

Because the output is redirected, DOS does not repair errors it encounters during the check; but it records all the errors in a report file. Afterward, you can use the CHKDSK command with the **/f** switch without redirection to correct any errors noted in the status report.

CHOICE

Used in batch files to display a specified prompt. The prompt provides you the opportunity to choose whether or not the batch program is to be processed.

Type

Batch, DOS, External

Syntax

choice [**/c**[:]*choices*][**/n**][**/s**] [**/t**[:]*c,nn*][*text*]

Parameters

text Specifies any text you want to be displayed before the prompt. Quotation marks around the text are optional. If you do not specify text, only the prompt is displayed.

Switches

/c[:]*choices*
Specifies allowable keys in a prompt. When displayed, the keys are separated by commas, appear in brackets ([]), and are followed by a question mark. If you do not specify the **/c** switch, the default of **yn** (for yes or no) is used. The colon (:) is optional.

/n Specifies that only the key selection be displayed. The text before the prompt is not displayed.

/s Specifies that the Choice program be case-sensitive. If the **/s** switch is not specified, the Choice program will accept either upper or lower case for any of the keys that you select.

/t[:]*c,nn*

Specifies a time out. After a specified period of time, a key that you specified with the **/c** switch is displayed as the default. The colon (:) is optional. The values for the **/t** switch are as follows:

c Specifies one of the keys you chose with the **/c** switch as the default.

nn Specifies the number of seconds to pause. Acceptable values are from 0 to 99.

Notes

The first key you assign with the **/c** switch returns a value of 1, the second a value of 2, the third a value of 3, and so on. If you press a key which is not among the keys you assigned, a warning beep is sounded (that is, it sends a BEL or 07H, character to the console).

If an error condition is detected, an *errorlevel* value of 255 is returned. If you press **Ctrl+Break**, an *errorlevel* value of 0 is returned.

Examples

What you see when you use CHOICE in a batch file

If you type the following syntax in a batch file,

```
choice /c:ync
```

you see the following message displayed when the batch program is started:

```
[Y,N,C]?
```

If you add text to the syntax,

```
choice /c:ync Yes, No, or Continue
```

the following is displayed when the batch program is started:

```
Yes, No, or Continue[Y,N,C]?
```

What you see if you leave out a prompt

If, as in the following example, you use the **/n** switch to leave out the prompt in a batch program,

```
choice /n Yes, No, or Continue?
```

only the text you specified is displayed when the batch program is started:

```
Yes, No, or Continue?
```

What you see if you use the /t switch

If you use the **/t** switch in a batch program, such as follows:

```
choice /c:ync /t:n,5
```

only the following is displayed when the batch program is started:

```
[Y,N,C]?
```

If, after 5 seconds, you have not pressed a key, the N key is displayed as the default and an *errorlevel* value of 2 is returned. If you press a key before 5 seconds elapse, the batch program is processed in accordance with your selection.

Using CHOICE in a batch program

The following batch program gives you the option of copying each of the following files: AUTOEXEC.BAT and CONFIG.SYS. The program copies the files from drive C to a BACKUP directory on drive P.

```
echo off
choice /c:yn Do you want to copy autoexec.bat?
if errorlevel 2 goto config
if errorlevel 1 goto doauto
:doauto
echo on
copy autoexec.bat autoexec.bak
echo off
:config
choice /c:yn Do you want to copy config.sys?
if errorlevel 2 goto exit
if errorlevel 1 goto doconfig
:doconfig
echo on
copy config.sys config.bak
echo off
:exit
```

CLS

Clears the screen.

The cleared screen shows only the command prompt and cursor.

For examples of using CLS in a batch-program menu system, see the *PC DOS User's Guide*.

Type

DOS, Internal, Network

Syntax

cls

CMOSCLK.SYS

This is a device driver that replaces the default DOS clock so that any request for the current date and time accesses CMOSCLK.SYS instead of the DOS system clock. It is normally used only if your current DOS system clock is not keeping the correct date.

Related Information

For detailed information, see "CMOSCLK.SYS" on page 253.

COMMAND

Starts a new instance of the DOS command interpreter, COMMAND.COM.

A *command interpreter* is a program that displays the command prompt at which you type commands. Use the EXIT command to stop the new command interpreter and return control to the old one.

Type

DOS, External

Syntax

command [[*drive*:]*path*][*device*][**/e:***nnnnn*][**/k:***filename*][**/p**] [**/c** *string*] [**/msg**]

In your CONFIG.SYS file, use the following syntax:

shell=[[*dos-drive*:]*dos-path*]**command.com** [[*drive*:]*path*] [*device*] **/e:***nnnn* **/p**

Parameters

[***drive***:]***path***
Specifies where the command interpreter is to look for the COMMAND.COM file when the transient part of the program needs to be reloaded. This parameter must be included if COMMAND.COM is not located in the root directory. This parameter is used to set the *comspec* environment variable.

device Specifies a different device for command input and output.

[***dos-drive***:]***dos-path***
Specifies the location of COMMAND.COM.

Switches

/**e:***nnnnn*

Specifies the environment size, where *nnnnn* is the size in bytes. The value of *nnnnn* must be in the range 160 through 32768. DOS rounds this number up to a multiple of 16 bytes. The default value is 256.

/**k:***filename*

Runs the specified program or batch file and then displays the DOS command prompt.

CAUTION:
Do not use this switch on the SHELL= statement line in you CONFIG.SYS file. Doing so can cause problems with applications and installation programs that make changes to your AUTOEXEC.BAT file.

/**p** Should be used only when the COMMAND.COM statement is used with the SHELL= statement in the CONFIG.SYS file. The /**p** switch makes the new copy of the command interpreter permanent. In this case, the EXIT command cannot be used to stop the command interpreter. If you specify /**p**, DOS runs your AUTOEXEC.BAT batch program when it carries out the corresponding SHELL= statement.

/**c** *string*

Specifies that the command interpreter is to perform the command specified by *string* and then stop.

/**msg**

Specifies that all error messages should be stored in memory. Usually, some messages are stored only on disk. This switch is useful only if you are running DOS from diskettes. You must specify the /**p** switch when you use the /**msg** switch. For more information about using the /**msg** switch, see the following "Notes" section.

Notes

Limits on environment size

If *nnnnn* is less than 160 or greater than 32768, DOS uses the default value of 256 bytes and displays the following message:

```
Parameter value not in allowed range.
```

Changing your terminal device

You can specify a different device (such as AUX) for input and output by using the *device* parameter. For more information about *device*, see command "CTTY" on page 59.

Running multiple command interpreters

When you start a new command interpreter, DOS creates a new command environment. This new environment is a copy of the parent environment. You can change the new environment without affecting the old one. The default size of the new environment is 256 bytes or the size of the current environment rounded up to the next 16 bytes, whichever is larger. Use the /**e** switch to override the default size.

Note: The current environment refers to the memory actually being used, not to the environment size specified with the previous **/e** switch.

Transient and resident memory

DOS loads the command interpreter into memory in two parts: the transient part (in memory) and the resident part (on disk). Some programs write over the transient part of COMMAND.COM when they run. When this happens, the resident part must locate the COMMAND.COM file on disk to reload the transient part. The *comspec* environment variable identifies where the COMMAND.COM file is located on the disk. If *comspec* is set to a diskette drive, DOS might prompt you to insert a diskette that contains the COMMAND.COM file.

Using the /msg switch

Usually, DOS keeps many error messages in the resident part of COMMAND.COM instead of using memory to store them. When DOS needs to display one of these messages, it retrieves the message from the disk containing the COMMAND.COM file.

If you are running DOS from diskettes instead of from a hard disk, DOS cannot retrieve such error messages unless you have the diskette containing the COMMAND.COM file in drive A. If this diskette is not present, DOS displays one of the following short messages instead of the full message:

```
Parse error

Extended error
```

You can make sure DOS displays complete error messages by using the **/msg** switch with the COMMAND command. This switch forces DOS to keep these error messages in memory so that they are always available when needed.

Use the **/msg** switch with the COMMAND command if you have a diskette system, unless you cannot afford to lose the memory used to store the error messages.

You must also specify the **/p** switch when you use the **/msg** switch.

Examples

The following command specifies that the DOS command interpreter is to start a new command interpreter from the current program, run a batch program named MYBAT.BAT, and then return to the first command interpreter:

```
command /c mybat.bat
```

The following command specifies that COMMAND.COM is located in the DOS directory on drive C:

```
c:\dos\command.com /e:1024
```

Since the full path for the COMMAND.COM file is specified, DOS sets the *comspec* environment variable to C:\DOS\COMMAND.COM. This command also creates an environment of 1024 bytes for this command interpreter.

Related Information

The SHELL command is the preferred method of using COMMAND to permanently increase space for the environment table. For more information about this alternative, see the command "SHELL" on page 214.

[COMMON]

The [COMMON] command is a block header for a configuration block. A configuration block is a set of CONFIG.SYS commands that you want DOS to run when that particular configuration is selected from the startup menu.

Type

CONFIG.SYS

Syntax

[COMMON]

Notes

Running several [COMMON] blocks

You can have as many [COMMON] blocks as you want. DOS runs the [COMMON] commands in the order in which they appear in the CONFIG.SYS file.

Recommended usage

Because some applications append commands to your CONFIG.SYS file, it is a good idea to place a [COMMON] command at the end of your [COMMON] block even though you do not add any commands to it.

Examples

The following CONFIG.SYS file defines two configurations and includes several commands that are common to both:

```
:
[COMMON]
files=30
buffers=30
lastdrive=z
break=on
device= C:\DOS\HIMEM.SYS
dos=HIGH,UMB
device=C:\DOS\EMM386.EXE NOEMS
device=C:\DOS\SETVER.EXE
```

```
[CPSW]
country=001,,C:\DOS\COUNTRY.SYS
devicehigh=C:\DOS\display.sys con=(ega,,1)

[DLS]
devicehigh=C:\NET\protman.dos /i:C:\NET
devicehigh=C:\NET\dlshelp.sys
devicehigh=C:\NET\ibmtok.dos

[INTLNK]
devicehigh=C:\DOS\INTERLNK.EXE

[COMMON]
devicehigh=C:\DOS\ANSI.SYS
shell=C:\DOS\COMMAND.COM /P /E:512
⋮
```

This CONFIG.SYS file configures the computer for LAN networking [DLS], laptop computer connectivity [INTLNK], and code page switching and keyboard support [CPSW]. For all three configurations, DOS runs the commands in the [Common] configuration blocks.

The INTLNK configuration uses the Client Device Driver (INTERLNK.EXE) to make the client computer use devices on the server computer as though they were local drives. INTERLNK loads itself into upper memory when upper memory blocks are available. The [Common] section makes these upper memory blocks available.

Related Information

For information about the INTERLNK.EXE device driver see Chapter 5, “Working With Device Drivers” on page 245

For more information about configuration blocks see the *PC DOS User's Guide*.

COMP

Compares the contents of two files or sets of files, byte by byte.

COMP can compare files on the same drive or on different drives, in the same directory or in different directories. As COMP compares the files, it displays their locations and filenames.

Type

DOS, External, Network

Syntax

comp [*data1*] [*data2*] [**/d**] [**/a**] [**/l**] [**/n=***number*] [**/c**]

Parameters

data1 Specifies the location and name of the first file or set of files you want to compare. You can use wildcards (* and ?) to specify multiple files.

data2 Specifies the location and name of the second file or set of files you want to compare. You can use wildcards (* and ?) to specify multiple files.

Switches

/**d** Displays differences in decimal format. (The default format is hexadecimal.)

/**a** Displays differences as characters.

/**l** Displays the number of the line on which a difference occurs, instead of displaying the byte offset.

/**n**=*number* Compares the first *number* of lines of both files, even if the files are different sizes.

/**c** Performs a comparison that is not case-sensitive.

Notes

Comparing files with the same names

The files you want to compare can have the same file name, provided they are in different directories or on different drives. If you do not specify a file name for *data2*, the default file name for *data2* is the same as the file name in *data1*. You can use wildcards (* and ?) to specify file names.

Special cases for *data1* and *data2*

If you omit necessary components of either *data1* or *data2* or if you omit *data2*, the Comp program prompts you for the missing information. If *data1* contains only a drive letter or a directory name with no filename, the default file name for *data1* is *.*. Therefore, all the files are compared in the specified directory to the file specified in *data2*. If *data2* contains only a drive letter or a directory name, the default file name for *data2* is the same as that in *data1*.

How the COMP command identifies mismatching information

During the comparison, messages are displayed to identify the locations of unequal information in the two files. Each message indicates the offset memory address of the unequal bytes and the contents of the bytes themselves (in hexadecimal notation unless you specify the /**a** or /**d** switch). The message has the following format:

```
Compare error at OFFSET xxxxxxxx
file1 = xx
file2 = xx
```

After 10 unequal comparisons, the Comp program stops comparing the files and displays the following message:

```
10 Mismatches - ending compare
```

Comparing files of different sizes

You cannot compare files of different sizes unless you specify the **/n** switch. If the file sizes are different, the following message is displayed:

```
Files are different sizes
Compare more files (Y/N)?
```

Press the **Y** key to compare another pair of files. Press **N** to stop the COMP command.

If you press **Y** in response to the prompt, the switches you specified on the command line are included in every comparison, until you press the **N** key or retype the command.

When comparing files of different sizes, use the **/n** switch to compare only the first portion of each file.

Comparing files sequentially

If you use wildcards to specify multiple files, the first file matching *data1* is found and compared with the corresponding file in *data2*, if it exists, the results of the comparison are displayed on the screen. The same is done for each file matching *data1*. When the comparison is finished the following message is displayed:

```
Compare more files (Y/N)?
```

To compare more files, press **Y**. You are prompted for the locations and names of the new files plus the switches to be used. If you do not specify any switches, the ones you specified previously are used. To stop the comparisons, press **N**.

If COMP cannot find the files

If the file or files you specify cannot be found, you are prompted with a message to determine whether you want to compare more files.

Related Information

For information about comparing the contents of two diskettes, see command "DISKCOMP" on page 76.

For information about doing a complete comparison of two files of different sizes, see command "FC" on page 105.

COPY

Copies one or more files to another location.

This command can also be used to combine files. When more than one file is copied, DOS displays each file name as the file is copied.

Type

DOS, Internal, Network

Syntax

copy [/**a**|/**b**] *source* [/**a**|/**b**][+ source [/**a**|/**b**][+ ...]][*destination*[/**a**|/**b**]][/**v**]

Note: The + is used when you want to combine files.

Parameters

source Specifies the location and name of a file or set of files from which you want to copy. *Source* can consist of a drive letter and colon, a directory name, a file name, or a combination.

destination

Specifies the location and name of a file or set of files to which you want to copy. *Destination* can consist of a drive letter and colon, a directory name, a file name, or a combination.

Switches

/a Indicates an ASCII text file. When the /**a** switch precedes the list of filenames (separated by plus signs) on the command line, it applies to all files whose names follow the /**a** switch, until the /**b** switch is encountered. In this case, the /**b** switch applies to the file whose name precedes the /**b** switch.

When the /**a** switch follows a filename, it applies to the file whose name precedes the /**a** switch and to all files whose names follow the /**a** switch, until the /**b** switch is encountered. The /**b** switch applies to the file whose name precedes the /**b** switch.

An ASCII text file can use an end-of-file character (**Ctrl+Z**) to indicate the end of the file. When using the COPY command to combine files, the files are treated as ASCII text files by default.

/b Indicates a binary file. When the /**b** switch precedes the list of file names on the command line, it applies to all files whose names follow the /**b** switch, until COPY encounters an /**a** switch. In this case, the /**a** switch applies to the file whose name precedes the /**a** switch.

When the /**b** switch follows a file name, it applies to the file whose name precedes the /**b** switch and to all files whose names follow the /**b** switch, until the /**a** switch is encountered. In this case, the /**a** switch applies to the file whose name precedes the /**a** switch.

The **/b** switch specifies that the command interpreter read the number of bytes specified by the file size in the directory. The **/b** switch is the default value for the COPY command unless you are using the COPY command to combine files (in which case the **/a** switch is the default).

/v Verifies that new files are written correctly.

Notes

Copying to and from devices

You can substitute a device name for one or more occurrences of *source* or for *destination*.

Using or omitting the /b switch when copying to a device

When *destination* is a device (for example, COM1 or LPT1), the **/b** switch causes DOS to copy data to the device in binary mode. In binary mode, all characters (including such special characters as **Ctrl+C**, **Ctrl+S**, **Ctrl+Z**, and carriage return) are copied to the device as data. Omission of the **/b** switch causes DOS to copy data to the device in ASCII mode. In ASCII mode, such special characters as those previously listed may cause DOS to take special action during the copying process.

Using the default destination file

If you do not specify a destination file, DOS creates a copy with the same name, creation date, and creation time as the original file, placing the new copy in the current directory on the current drive. If the source file is on the current drive and in the current directory and you do not specify a different drive or directory for the destination file, the COPY command stops and DOS displays the following error message:

```
File cannot be copied onto itself
0 File(s) copied
```

Using the /v switch

If DOS cannot verify a write operation, it displays an error message. Although recording errors rarely occur with the COPY command, the **/v** switch lets you verify that critical data has been correctly recorded. The */v* switch also slows down the COPY command, because DOS must check each sector recorded on the disk.

Using the /a and /b switches

The effect of an **/a** or **/b** switch depends upon its position on the command line. When the **/a** or **/b** switch follows the source file name, the COPY command performs as shown in the following list:

/a Treats the file as an ASCII (text) file and copies data that precedes the first end-of-file character. The first end-of-file character and the remainder of the file is not copied.

/b Copies the entire file, including any end-of-file character.

When the **/a** or **/b** switch follows the destination file name, the COPY command performs as shown in the following list:

/a Adds an end-of-file character as the last character of the file.

/b Does not add an end-of-file character.

Combining files with the COPY command

If you specify more than one *source*, separating entries with a plus sign (+), will combine the files, creating a single file. If you use wildcards in *source* but specify a single file name in *destination*, all the files matching the file name in *source* are combined. A single file is created with the file name specified in *destination*.

In either case, the combined files are assumed to be ASCII files unless you specify the **/b** switch.

If the name of the destination file is the same as the name of one of the files being copied (except the first file), the original contents of the destination file are lost. When this happens, the following message is displayed:

```
Content of destination lost before copy
```

Copying files in subdirectories

To copy all files and subdirectories of a directory, you should use the XCOPY command.

Copying zero-length files

The COPY command does not work with files that are 0 bytes long. Use XCOPY to copy these files.

Changing the time and date of a file

If you want to assign the current time and date to a file without modifying the file, use the following format. The commas indicate the omission of the *destination* parameter.

```
copy /b source+,,
```

Examples

The following command copies a file and ensures that an end-of-file character is at the end of the copied file:

```
copy memo.doc letter.doc /a
```

To copy a file named ROBIN.TYP from the current drive and directory to an existing directory named BIRDS that is located on drive C, type the following command:

```
copy robin.typ c:\birds
```

If the BIRDS directory does not exist, DOS copies the file ROBIN.TYP into a file named BIRDS that is located in the root directory on the disk in drive C.

To copy several files into one file, list any number of files as *source* parameters on the COPY command line. Separate file names with a plus sign (+) and specify a file name for the resulting combined file, as the following example shows:

```
copy mar93.rpt + apr93.rpt + may93.rpt report
```

This command combines the files named MAR93.RPT, APR93.RPT, and MAY93.RPT from the current drive and directory and places them in a file named REPORT in the current directory on the current drive. When files are combined, the destination file is created with the current date and time. If you omit *destination*, DOS combines the files and stores them under the name of the first specified file. For example, if a file named REPORT already exists, you can use the following command to combine all four files in REPORT:

```
copy report + mar93.rpt + apr93.rpt + may93.rpt
```

You can also combine several files into one by using wildcards, as the following example shows:

```
copy *.txt combin.doc
```

This command combines all files in the current directory on the current drive that have the extension .TXT into one file named COMBIN.DOC, also in the current directory on the current drive.

If you want to combine several binary files into one by using wildcards, include the **/b** switch, as the following example shows:

```
copy /b *.exe combin.exe
```

This prevents DOS from treating **Ctrl+Z** as an end-of-file character.

CAUTION:
If you combine binary files, the resulting file might not be usable due to internal formatting.

In the following example, the COPY command combines each file that has a .TXT extension with its corresponding .REF file. The result is a file with the same file name but with a .DOC extension. Thus, FILE1.TXT is combined with FILE1.REF to form FILE1.DOC. Then FILE2.TXT is combined with FILE2.REF to form FILE2.DOC, and so on.

```
copy *.txt + *.ref *.doc
```

The following COPY command combines all files with the .TXT extension, then all files with the .REF extension into one file named COMBIN.DOC:

```
copy *.txt + *.ref combin.doc
```

Related Information

For information about copying directories and subdirectories, see command "XCOPY" on page 240.

COUNTRY

Enables DOS to use international time, dates, currency, case conversions, and decimal separators.

The COUNTRY command configures DOS to recognize the character set and punctuation conventions observed when using one of the supported languages.

Type

CONFIG.SYS

Syntax

country=*xxx*[,[*yyy*][,[*drive*:][*path*]*filename*]]

Parameters

xxx Specifies the country code.

yyy Specifies the code page for the country.

[***drive***:][***path***]***filename***
Specifies the location and name of the file containing country information.

Notes

Changing default settings

DOS uses the United States as the default setting. You can use the COUNTRY command in your CONFIG.SYS file to change the setting.

If you do not specify the location and name of the file containing country-specific information, DOS tries to find the COUNTRY.SYS file in the root directory of your startup drive.

Specifying country or language support

The following table lists the country code, code pages, and the time and date formats for each country or language supported by DOS. The country code specifies the time and date format for the supported country or language, and the code page specifies the character set used to display characters on the screen or printer.

If you use country code 003, you can use only code page 437 or 850 for the *yyy* parameter. The first of the two code pages listed for each country or language is its default code page. In the table, for each country code, the "Date format" column shows how DOS displays January 19, 1993, and the "Time format" column shows how DOS displays 5:35 P.M. (with 0 seconds and 0 hundredths of a second).

Country or language	Country code	Code pages	Date format	Time format
United States	001	437, 850	01/19/1993	5:35:00.00p
Canadian-French	002	850, 863	1993-01-19	17:35:00,00
Latin America	003	850, 437	19/01/1993	5:35:00.00p
Russsia	007	866, 855 (Russian DOS only)	19/01/1993	17:35:00,00
Netherlands	031	850, 437	19-01-1993	17:35:00,00
Belgium	032	850, 437	19/01/1993	17:35:00,00
France	033	850, 437	19.01.1993	17:35:00,00
Spain	034	850, 437	19/01/1993	17:35:00,00
Bulgaria	035	855, 850	19.01.1993	17:35:00,00
Hungary	036	852, 850	1993-01-19	17:35:00,00
Bosnia	038	852, 850	1993-01-19	17:35:00,00
Yugoslavia	038	852, 850	1993-01-19	17:35:00,00
Italy	039	437, 850	19/01/1993	17.35.00,00
Romania	040	852, 850	1993-01-19	17:35:00,00
Switzerland	041	850, 437	19.01.1993	17,35,00.00
Czech	042	852, 850	1993-01-19	17:35:00,00
Slovakia	042	852, 850	1993-01-19	17:35:00,00
United Kingdom	044	850, 437	19/01/1993	17:35:00.00
Denmark	045	850	19-01-1993	17.35.00,00
Sweden	046	850, 437	1993-01-19	17.35.00,00
Norway	047	850	19.01.1993	17:35:00,00
Poland	048	852, 850	1993-01-19	17:35:00,00
Germany	049	850, 437	19.01.1993	17:35:00,00
Brazil	055	850, 437	19/01/1993	17:35:00,00
International English	061	850, 437	19/01/1993	17:35:00.00
Portugal	351	850, 860	19-01-1993	17:35:00,00
Albania	355	852, 850	19.01.1993	17:35:00,00
Finland	358	850, 437	19.01.1993	17:35:00,00
Serbia and Montenegro	381	855, 852	19.01.1993	17:35:00,00
Croatia	384	852, 850	19.01.1993	17:35:00,00
Slovenia	386	852, 850	19.01.1993	17:35:00,00
FYR Macedonia	389	855, 852	19.01.1993	17:35:00,00

Code pages for the following countries or languages are also available with special versions of DOS: Arabic, Israel, Japan, Korea, People's Republic of China, and Taiwan.

Examples

To convert international currency, time, date, and case to French conventions, add the following command to your CONFIG.SYS file:

```
country=033
```

For this example, assume that the COUNTRY.SYS file is in the root directory of the startup drive. If COUNTRY.SYS is in a different location, you specify the location in [*drive*:]*path* on the command line.

To specify a code page with the country code for France, type the following:

```
country=033,850
```

If you omit the code page but include the [*drive*:][*path*]*filename* parameter, you must still type the comma that would have preceded the code page, as the following example shows:

```
country=033,,c:\dos\country.sys
```

Related Information

For information about changing characters and their arrangement on your keyboard, see command "KEYB" on page 138.

For information about preparing and selecting code pages, see command "MODE" on page 158.

For information about loading country-specific information, see command "NLSFUNC" on page 173.

CPBACKUP

Makes a back up copy of data to diskettes, tape, or a network drive. CPBACKUP is a full-screen utility that can backup files using full, incremental, differential, or unattended methods. You can compress, encrypt, and virus- check data as it is backed up.

Type

DOS, External

Syntax

cpbackup [*d:*][*setname*][*filespec...*][/date=*mmddyy-mmddyy*][**/exattr=***hsr*] [**/full**|**/inc**|**/sep**|**/copy**|**/fullerase**|**/dif**][**/drive=x** | **/drive=***d:n*] [**/addr=***base-i-d*][**/rate=***rate*][**/no**] [**/r**[*setup file name*|*filespec*]][**/ecc**|**/noecc**][**/save**|**/nosave**][**/sf**|**/nosf**] [**/mtask**][**/?**][**/video**][/*video options*]

Note: You can use as many options as you need. However, combinations that affect the same option (such as **/full** and **/sep**) are not allowed together. If two mutually exclusive options are specified, the last one on the command line takes effect and the first is ignored.

Parameters

d: Specifies a setup drive to back up or restore to. This overrides any drive that is specified in the CPBACKUP.CFG file and the setup file.

setname Loads the specified setup file. When CPBACKUP is started with a setup file, the directory tree and file list are always visible.

filespec... Accepts any valid DOS filespec (specification for name and location of file or files), including paths and wildcards. For example, to back up all *.DAT files in your \WORK directory, type `cpbackup\work\*.dat` at the DOS prompt. You can also add file specifications to a specified setup file at the command line. See "Adding file specifications to setup file from DOS prompt" under the "**Examples**" heading.

Switches

Switches used with file selection

/date=*mmddyy-mmddyy*
Specifies dates of files to back up.

/exattr=*hsr*
Exclude (h)idden, (s)ystem, (r)ead-only.

/full Specifies a full backup is marked as being backed up.

/copy Specifies a full backup. Do not mark as backed up.

/fullerase
Specifies a full backup after the tape has been erased.

/inc Specifies an incremental backup, and append to the full backup.

/sep Specifies a separate incremental backup.

/dif Specifies a differential backup.

Switches used with hardware selections

/drive=x | **/drive=***d:n*
The **/drive=x** switch specifies the word *tape*. The **/drive=***p:n* switch specifies the media and drive size to back up to (360, 720, 1200, or 1400).

/addr=*base-i-d*
Specifies I/O addresses for a tape drive.

base Specifies the hexidecimal base address.

i Specifies the interrupt request (IRQ).

d Specifies the direct memory access (DMA) channel.

/rate=*rate*
Sets the data rate (rate=1000KB per second, 500KB per second or 250KB per second - controller must support).

/no Specifies not to use overlapped input and output. Use of this switch turns off the use of simultaneous hard disk and diskette DMA.

Switches used for general selections.

/r Specifies that the restore mode is to start automatically. This switch starts the backup in the restore mode and immediately prompts you to insert the last diskette (if using diskettes) or tape (if using tapes) of the backup set you want to restore. The directory is read and the restore automatically begins, using the specifications in the .CFG file.

/r *setup file name*
Starts backup in restore mode and loads the settings that are saved in the specified setup file. This includes information such as the current drive and all files listed in a Include and Exclude Files dialog. You must specify this parameter before any others to be effective.

/r *filespec*
The filespec parameter with /r restores specific files. You are prompted to insert the media containing the directory of the backup set, then prompted to insert the disk or tape containing the files matching your file specifications. For example, CPBACKUP /R*.WKS restores only your Lotus 1-2-3** data files.

/save Specifies that the history is to be saved to the hard disk.

/nosave Specifies that the history is not to be saved to the hard disk.

/ecc Specifies that error correction is to be used.

/noecc Spcifies that error correction is not to be used.

/sf Specifies that standard formatting is to be used.

/nonsf Specifies that nonstandard formatting is to be used.

/mtask Specifies protection against files being changed in multitasking modes.

/? Displays help for the command line options.

/video Displays help for the following video and mouse selections:

/in Specifies that you use the default color scheme instead of the configured scheme.

/bw Specifies that you use a black and white color scheme (for a black and white display screen).

/mono Specifies the use of the monochrome color scheme (IBM monochrome).

/lcd Uses a LCD color scheme (for laptops).

/ff Speeds up the display. This could cause a snow affect on some display devices (CGA only).

/bf Specifies the BIOS font. This is used if graphics do not display properly.

/nf No fonts - Specifies that graphics characters are not to be used.

/ngm No graphics mouse pointer - Specifies that a character mouse pointer is to be used.

/le Specifies a left-handed mouse (exchange of left and right mouse buttons functions).

/im Ignore mouse - Specifes that the mouse is not to be used in this program.

/ps2 Resets the mouse hardware on a PS/2*. (Use if the mouse disappears or freezes.)

Examples

Specifying the drive to use

To specify drive E, type:

```
cpbackup e
```

Adding file specifications to setup file from DOS prompt

To add all *.DBF and *.WKS files to those already specified in a setup file called DAILY, type:

```
cpbackup daily*.dbf*.wks
```

Running CPBACKUP from a batch file

You can select your backup options and then save them so they can be run from a batch file, or you can create a batch file using the command-line options described earlier, to run the program.

For example, you can use the preconfigured setup file SPREAD to back up all of your Lotus 1-2-3 files. Create a batch file with the following line:

```
cpbackup spread
```

This command starts the backup, loads in all the settings as saved in SPREAD, and backs up all your Lotus files. When the backup is complete, the program returns to the next command in the batch file.

When the backup is started with a setup file from a batch file, express is disabled. This means the directory tree and file list are always visible.

Another example of using command-line options is to add the following line to a batch file:

```
cpbackup /bw /no /full /rate=1
```

This command runs the backup in black and white mode at medium speed, backs up all directories and files, and sets the tape controller card at 1,000KB per second. When the backup is complete, the next command in the batch file appears.

Note: If the backup is started from a batch file, and the backup is completed without interruption, control is automatically returned to the batch file to execute the next command. Otherwise, if the backup process is interrupted (such as responding to a dialog box or inadvertently pressing keys), control cannot be returned to the batch file until you exit from the program.

Scheduling Unattended Backups

The Scheduler button and the Schedule Backups commands use the Scheduler, which is an application that lets you schedule backups without monitoring the procedure. This is especially useful for backing up to tape drives, removable cartridges, or to a network volume.

CPBDIR

Starts the CPBDIR program. This program determines the number of diskettes and the correct order of a high-speed or medium-speed diskette backup, as well as give information about how the backup was made.

Note: For low-speed backups, use the DOS DIR command to find out the disk number and date of the backup. There will be two files on the disk: CPBACKUP.INF and CPBACKUP:*nnn*, where *nnn* is the disk number of the set. There will also be a CPB*xx*.DIR on the last disk of the set (this is the history file).

Type

DOS, External

Syntax

cpbdir *drive*:[/**x**]

Parameters

drive: Drive letter of diskette drive containing the CPBACKUP diskette to view.

Switches

/x Displays extended information about the diskette.

Notes

The CPBDIR command displays information about the CPBACKUP diskette, such as the diskette number of a set, media used, formatting, speed used, and if a directory exists on the diskette.

CPSCHED

The CPSCHED comamnd activates CPSCHED.EXE which is the memory resident portion of the Scheduler program. You can add the CPSCHED statement to your AUTOEXEC.BAT file so that it automatically loads each time you turn on your computer. The Scheduler program schedules other DOS programs to run automatically at a preset date and time. Scheduling is especially useful for lengthy procedures that can run unattended.

If CPSCHED is already started and you want to add or edit scheduled events use, the SCHEDULE command.

Type

DOS, External

Syntax

cpsched [/**u**][/**low**]

Switches

/u	Specifies that the currently loaded resident version of CPSCHED be removed from memory.
/low	Specifies that CPSCHED not be loaded into the upper memory blocks.

Notes

CPSCHED provides a 15-second warning that a scheduled program is about to run. When the program is complete, the scheduler returns to the application you were running.

CPSCHED must be resident in order to launch the program at the scheduled time.

Examples

To load CPSCHED, from DOS type:

```
cpsched
```

To remove CPSCHED from memory, type:

```
cpsched /u
```

Related Information

For related information see command "SCHEDULE" on page 205.

CTTY

Changes the terminal device used to control your system.

Use the CTTY command if you want to use another device to enter commands.

Type

DOS, Internal, Network

Syntax

ctty *device*

Parameters

device Specifies the alternative device you want to use to type DOS commands.

Notes

Using valid values for *device*

The valid values for the *device* parameter are: *prn*, *lpt1*, *lpt2*, *lpt3*, *con*, *aux*, *com1*, *com2*, *com3*, *com4*.

Setting up the serial port for CTTY

Use the MODE command to set up your serial port for baud rate, parity, bits, and stop bit before using the CTTY command.

Using CTTY with programs that do not use DOS

Many programs do not use DOS for input or output. These programs send input directly to the hardware on your computer. The CTTY command has no effect on these programs; it affects only programs that use DOS for reading keyboard input and displaying output.

Setting the terminal device with command

In addition to the CTTY command, you can use the *device* parameter of the COMMAND command to specify the input device.

Examples

The following command changes control of all input and output from the current device (your computer screen and keyboard) to the AUX port:

```
   ctty aux
```

In this example, a remote terminal device connected to the AUX port controls input and output for your system.

To transfer input and output back to the screen and keyboard, type the following command at the remote terminal:

```
    ctty con
```

Related Information

For more information about changing the input device when specifying a command interpreter, see command "COMMAND" on page 40.

For more information about setting up the serial port, see command "MODE" on page 158.

DATAMON

Starts the Data Monitor program giving you two options (Delete Tracker or Delete Sentry**) to track deleted files. Delete Tracker saves the name of each deleted file and a list of the cluster numbers it occupied in a hidden file called PCTRACKR.DEL. Delete Sentry saves each deleted file in a hidden directory called \SENTRY. Both of these tracking methods provide a means to recover deleted files by using the UNDELETE command as soon after a deletion as possible.

You cannot use both options at the same time. You need to add DATAMON with one of the options to your AUTOEXEC.BAT file. This will ensure your use of the same tracking system each time you start your computer. However, should you want or need to disable the option temporarily, you can do so from the command prompt by typing one of the following commands:

```
    datamon /sentry-
    datamon /tracker-
```

Type

DOS, External

Syntax

datamon [/load][/low][/u][/s][/sentry+][/sentry-][/tracker+][/tracker-]

Switches

/load Loads the Data Monitor program using the options in the DATAMON.INI file.

/low Loads the Data Monitor program into conventional memory even if upper memory is available.

/u Removes the Data Monitor program from your computers memory.

/s Indicates which tracking option is active in the Data Monitor program.

/sentry+	Activates the Delete Sentry option.
/sentry-	De-activates the Delete Sentry option.
/tracker+	Activates the Delete Tracker option.
/tracker-	De-activates the Delete Tracker option.

DATE

Displays the date or allows you to change the date from your terminal or from a batch program.

DOS records the current date for each file you create or change; this date is listed next to the file name in the directory.

Type

DOS, Internal, Network

Syntax

date [*mm-dd-yy*]

Parameters

mm-dd-yy

Sets the date you specify. Values for day, month, and year must be separated by periods (.), hyphens (-), or slash marks (/). The date format depends on the COUNTRY setting you are using in your CONFIG.SYS file. The following list shows the valid values for the month, day, and year portions of the *mm-dd-yy* parameter.

mm	1 through 12
dd	1 through 31
yy	80 through 99 or 1980 through 2099

Notes

Adjusting for days in a month

DOS is programmed to change months and years correctly, whether the month has 28, 29, 30, or 31 days.

Using the DATE command in your AUTOEXEC.BAT file

When you use an AUTOEXEC.BAT file, DOS does not automatically display a prompt for a date when you start your system. To prompt users for the date every time the system is restarted, include the DATE command in AUTOEXEC.BAT.

Changing the date format

It is possible to change the *mm-dd-yy* format to display the date in other formats. You can add the COUNTRY command to your CONFIG.SYS file to change the date format to the European standard (*dd-mm-yy*) or to the Scientific International (Metric) format (*yy-mm-dd*).

Related Information

For information about changing the current time, see "TIME" on page 228.

DEBUG

Starts Debug, a program that allows you to test and debug executable files. After typing debug at the command prompt, the Debug program prompt (a - hyphen) is displayed. All of the Debug program commands are typed after this prompt. To quit the Debug program and return to the DOS command prompt, type q at the Debug program prompt and press enter.

Type

DOS, External

Syntax

debug [[*drive*:][*path*]*filename* [*testfile-parameters*]]

Parameters

[***drive***:][***path***]***filename***
Specifies the location and name of the executable file you want to test.

testfile-parameters
Specifies any command-line information required by the executable file you want to test.

DEBUG commands

The following information is provided as a quick reference for the DEBUG commands.

Syntax	**Command**
?	Displays a list of the DEBUG commands when typed after the Debug program prompt.
A [*address*]	A (ASSEMBLE): Assembles IBM Personal Computer Macro Assembler language statements directly into memory.
C *range address*	C (COMPARE): Compares the contents of two portions of memory.
D [*range*]	D (DUMP): Displays the contents of a portion of memory.

Syntax	Command
E *address*	E (ENTER): Displays and allows modification of data in memory in a sequential manner, starting at the specified address.
E *address list*	E (ENTER): Replaces the contents of one or more bytes, starting at the specified address (*addr*) with the value specified by (*list*).
F *range list*	F (FILL): Fills a range of memory with values specified by list.
G	G (GO): Processes the program you are debugging without break points.
G [*=addr*] [*addresses*]	Stops the processing of the program when the instruction at the specified address (*=addr*) is reached (breakpoint), and displays the registers, flag, and the next instruction to be processed.
H *value1 value2*	H (HEX): Performs hexadecimal arithmetic.
I *port*	I (INPUT): Inputs and displays (in hexadecimal) one byte value from a specified port.
L [*addr*][*drive#*] [*startsec*][*numsec*]	L (LOAD): Loads the contents of a file or disk sectors into memory at a specified location.
M *range address*	M (MOVE): Moves the contents of the memory location specified by *range* to the location beginning at the address specified by *addr*.
N [*pathname*][*arglist*]	N (NAME): Specifies the pathname and a file for use with an L or W command, or specifies the parameters for the file you are testing.
O *port byte*	O (OUTPUT): Sends one byte value to the specified output port.
P [*addr*][*value*]	P (PROCEED): Executes a loop, a repeated string instruction, a software interrupt, a subroutine call, or traces through any other instruction.
Q	Q (QUIT): Ends the Debug program.
R	R (DISPLAY ALL): Displays the contents of all registers and flags and the next instruction to be processed.
R *f*	R (DISPLAY FLAGS): Displays all flags.
R *register*	R (DISPLAY REGISTER): Displays or alters the contents of one or more central-processing-unit (CPU) registers.
S *range list*	S (SEARCH): Searches the *range* for the characters specified by *list*.
T [*=address*] [*value*]	T (TRACE): Processes one or more instructions starting at CS:IP, or at *=addr*, if specified. It also displays the contents of all registers, the status of all flags, and the decoded form of the instruction that Debug will execute next.

Syntax	Command
U [*range*]	U (UNASSEMBLE): Disassembles bytes and displays the corresponding source statements.
W [*addr*][*drive#*] [*startsect*][*numsect*]	W (WRITE): Writes data to a specified disk sector at a specified location.
XA *count*	XA (ALLOCATE EXPANDED MEMORY): Allocates a specified number (*number*) of expanded memory pages to a handle.
XD *handle*	Removes expanded memory allocation.
XM *lpage ppage handle*	Maps an expanded memory specification (EMS) logical page to an EMS physical page from an EMS handle.
XS	XS (DISPLAY EXPANDED-MEMORY STATUS): Displays the status of expanded memory.

Notes

Specifying valid address entries

An *address* parameter in a DEBUG command specifies a location in memory. *Address* is a two-part designation containing either an alphabetic segment register or a 4-digit segment address, plus an offset value. You can omit the segment register or segment address. The default segment for the **A**, **G**, **L**, **T**, **U**, and **W** commands is CS. The default segment for all other commands is DS. All numeric values are in hexadecimal format.

The following are valid addresses:

```
CS:0100

04BA:0100
```

The colon between the segment name and the offset value is required.

Specifying valid range entries

A *range* parameter in a DEBUG command specifies a range of memory. You can choose from two formats for *range*: a starting address and an ending address, or a starting address and the length (denoted by *l*) of the range.

For example, both of the following syntaxes specify a 16-byte range beginning at CS:100:

```
cs:100 10f

cs:100 l 10
```

DEFRAG

Reorganizes files on a disk to optimize disk performance. There are two ways you can use the DEFRAG command. You can use it with the **/f** switch which allows you to fully optimize your disk. You are also able to specify a sort order for your files. The second and slightly faster method to reorganize your disk is to use the **/u** switch with the DEFRAG command. Your files are unfragmented leaving space between them. If you use the **/u** switch, you cannot use the sort order feature.

Type

DOS, External

Syntax

defrag [*drive*:][**/f**][**/sorder**][**/b**][**/skiphigh**][**/lcd**|**/bw**|**/g0**]

defrag [*drive*:][**/u**][**/b**][**/skiphigh**][**/lcd**|**/bw**|**/g0**]

Parameters

***drive*:** Specifies the drive that contains the disk you want to optimize.

Switches

/f Defragments files and ensures that the disk contains no empty spaces between files.

/u Defragments files and leaves empty spaces, if any, between files.

/sorder Controls how the files are sorted in their directories. If you omit this switch, the Defrag program uses the current order on the disk. The following list describes each of the values you can use to sort files. Use any combination of the values, and do not separate these values with spaces.

- ***n*** Arranges files in alphabetic order by name.
- ***-n*** Arranges files in reverse alphabetic order by name (Z through A).
- ***e*** Arranges files in alphabetic order by extension.
- ***-e*** Arranges file in reverse alphabetic order by extension (Z through A).
- ***d*** Arranges files by date and time, earliest first.
- ***-d*** Arranges file by date and time, latest first.
- ***s*** Arranges files by size, smallest first.
- ***-s*** Arranges files by size, largest first.

/b Restarts your computer after files have been reorganized.

/skiphigh Loads the Defrag program into conventional memory. By default, the Defrag program is loaded into high memory.

/lcd Runs the Defrag program using an LCD color scheme.

/bw Runs the Defrag program using a black and white color scheme.

/g0 Disables the graphic mouse and graphic character set.

Notes

Network and INTERLNK drives

You cannot use the Defrag program to optimize network drives or drives created with INTERLNK.

Disk information reported by the Defrag program

The Defrag program provides the following information:

- User files are reported as one number.
- The root file is counted as a directory.
- The volume label is not counted as a file.

This is in contrast to the disk information reported when you use the Chkdsk program. See the command "CHKDSK" on page 34 for information.

Start DEFRAG only from DOS

If you start the Defrag process from a program such as Windows, you might lose data.

DEFRAG exit codes

0 The process completed successfully.
1 An internal error occurred.
2 The disk does not contain any free clusters (to operate, DEFRAG needs 1 free cluster).
3 **Ctrl+C** was used to stop the process.
4 A general error occurred.
5 An error occurred while a cluster was being read.
6 An error occurred while a cluster was being written.
7 A disk allocation error occurred; run CHKDSK to correct.
8 A memory error occurred.
9 There is not enough memory to defragment the disk.

Examples

The following example specifies that DEFRAG be loaded into conventional memory and that DEFRAG reorder files on drive C by directory, sorting files according to the date they were created, from latest created to earliest created. This example fully optimizes drive C, but slows DEFRAG.

```
defrag c: /f /s:-d /skiphigh
```

DEL (ERASE)

Deletes specified files.

Type

DOS, Internal, Network

Syntax

del [*drive*:][*path*]*filename* [/**p**]

erase [*drive*:][*path*]*filename* [/**p**]

Parameters

[***drive***:][***path***]***filename***
Specifies the location and name of the file or set of files you want to delete.

Switches

/**p** Prompts you for confirmation before deleting the specified file.

Notes

Using the */p* switch

If you use the */p* switch, the name of the file being deleted and a confirmation prompt are displayed as follows:

```
filename, Delete (Y/N)?
```

Press **Y** to confirm the deletion, **N** to cancel the deletion and display the next filename (if you specified a group of files), or **Ctrl+C** to stop the DEL command.

Deleting more than one file at a time

You can delete all the files in a directory by typing the DEL command followed by [*drive*:]*path*. You can also use wildcards (* and ?) to delete more than one file at a time. However, you should use wildcards cautiously with the DEL command to avoid deleting files unintentionally. When you type the following command:

```
del *.*
```

the message prompt:

```
All files in directory will be deleted!
Are you sure (Y/N)?
```

is displayed.

Press **Y** and then **Enter** to delete all files in the current directory, or press **N** and then **Enter** to cancel the deletion.

Before you use wildcards with the DEL command to delete a group of files, you can use the same wildcards with the DIR command to see a list of the names of all the files included in the group.

Warning: Once you delete a file from your disk, you might not be able to retrieve it. Although the UNDELETE command can retrieve deleted files, it can do so with certainty only if no other files have been created or changed on the disk. If you accidentally delete a file that you want to keep, stop what you are doing and immediately use the UNDELETE command to retrieve the file.

Examples

To delete all the files in a directory named TEST on drive C, you can use either of the following commands:

```
del c:\test

del c:\test\*.*
```

Related Information

For information about retrieving a deleted file, see command "UNDELETE" on page 231.

For information about removing a directory, see command "RMDIR (RD)" on page 203

DELTREE

Deletes a directory and all the subdirectories and files in it.

Type

DOS, External

Syntax

deltree [/y][*drive*:]*path*[[*drive*:]*path*[...]]

Parameters

[*drive*:]*path*
Specifies the name of the directory you want to delete. You can specify as many directories as you want.

Switches

/y Suppresses prompting to confirm whether you want to delete the subdirectory.

Notes

Use the DELTREE command cautiously. Every file and subdirectory within the specified directory will be deleted.

DEVICE

Loads into memory the device driver you specify.

Type

CONFIG.SYS

Syntax

device=[*drive*:][*path*]*filename* [*dd-parameters*]

Parameters

[***drive***:][***path***]***filename***

Specifies the location and name of the device driver you want to load.

[***dd-parameters***]

Specifies any command-line information required by the device driver.

Notes

Using standard device drivers

There are several installable device drivers provided with DOS. Each one is individually described in detail in Chapter 5, "Working With Device Drivers" on page 245.

The files COUNTRY.SYS and KEYBOARD.SYS are not device drivers. DOS loads these files whenever necessary. Do not try to load either of these files with the DEVICE command. If you do, your system locks up and you cannot restart DOS. For information about loading COUNTRY.SYS, see the command "COUNTRY" on page 51. For information about loading KEYBOARD.SYS, see the command "KEYB" on page 138 .

Installing device drivers for other products

When you purchase a mouse, a scanner, or a similar product, the manufacturer usually includes device-driver software. To install a device driver, specify its location and name on a DEVICE command line.

Installing a third-party console driver

If you install both DISPLAY.SYS and a third-party console driver, such as VT52.SYS, the third-party device driver must be installed first. Otherwise, the third-party device driver might disable DISPLAY.SYS.

Examples

If you plan to use an ANSI escape sequence to control the screen and keyboard, you should add the following command to your CONFIG.SYS file (assuming DOS files are in the DOS directory on drive C):

```
device=c:\dos\ansi.sys
```

Related Information

For information about loading device drivers into the upper memory area, see the command "DEVICEHIGH."

DEVICEHIGH

Loads device drivers into the upper memory area.

Loading a device driver into the upper memory area frees more bytes of conventional memory for other programs. If upper memory is not available, the DEVICEHIGH command functions just like the DEVICE command.

Note: You can use the DEVICEHIGH command only in your CONFIG.SYS file.

Type

CONFIG.SYS

Syntax

devicehigh=[*drive*:][*path*]*filename* [*dd-parameters*]

To specify the regions in memory into which to load the device driver, use the following syntax:

devicehigh=[[/l:*region1*[,*minsize1*][;*region2*[,*minsize2*] [/**s**]] [*drive*:][*path*] *filename* [*dd-parameters*]

Parameters

[*drive*:][*path*]*filename*
Specifies the location and name of the device driver you want to load into the upper memory area.

dd-parameters
Specifies any command-line information required by the device driver.

Switches

/l:*region1*.[,*minsize1*][;*region2*[,*minsize2*]...

Specifies one or more regions of memory in which to load the device driver. By default, DOS loads the driver into the largest free upper-memory block (UMB) and makes all other UMBs available for use by the driver. You can use the **/l** switch to load the device driver into a specific region of memory, or you can use the switch to specify which regions the driver can use.

To load the driver into the largest block in a specific region of upper memory, specify the region number after the **/l** switch. For example, to load the driver into the largest free block in region 4, you would type

```
/l:4
```

. You can use the MEM **/f** command to list the free areas of memory.

When loaded with the **/l** switch, a device can use only the specified memory region. Some device drivers use more than one area of memory. For those drivers, you can specify more than one region. You can use MEM **/m** command (followed by the name of the module) to determine how a particular device driver uses memory. To specify two or more regions, separate the block numbers with a semicolon (;). For example, to use blocks 2 and 3 type:

```
  /l:2;3
```

Normally, DOS loads a driver into a UMB in the specified region only if that region contains a UMB larger than the load size of the driver (usually equal to the size of the executable program file). If the driver requires more memory while running than it did when it was loaded, you can use the *minsize* parameter to ensure that the driver is not loaded into a UMB that is too small for it. If you specify a value (must be in bytes) for *minsize*, DOS loads the driver into that region only if it contains a UMB that is larger than both the load size and the *minsize* value.

/S Shrinks the UMB to its minimum size while the driver is loading. Using this switch make the most efficient use of memory. This switch is used only in conjunction with the **/l** switch and it affects only UMBs for which a minimum size was specified.

Notes

If no upper memory area is available

If there is not enough upper memory area available to load the device driver you specified with the DEVICEHIGH command, DOS will load it into conventional memory (as if you had used the DEVICE command).

Affect of RAMBoost on the DEVICEHIGH command

If the RAMBoost program is running or if you start the RAMBoost program, all DEVICEHIGH commands are reconfigured. Programs are analyzed by RAMBoost and positioned where RAMBoost determines is best for your computer.

Examples

The following command statement in your CONFIG.SYS file directs DOS to run the MYDRVR.SYS driver in upper memory, and load the driver into upper memory block 2.

```
devicehigh=/l:2 c:\mydrvr.sys
```

The following command loads the same driver in the previous example into upper memory regions 1 and 3 (but only if each region is at least 25K in size).

```
devicehigh=/l:1,25;3,25 c:\tools\mydrvr.sys
```

Related Information

For information about loading programs into the upper memory area, see the command "LOADHIGH" on page 146

For information about loading device drivers into conventional memory, see the command "DEVICE" on page 69.

For information about the RAMBoost program, see "RAMSETUP" on page 194. Also, see the *PC DOS User's Guide*.

For information about device drivers, see Chapter 5, "Working With Device Drivers" on page 245.

DIR

Displays a list of files and subdirectories of a directory.

You can use the DIR command without parameters or switches. When you do, the following information is displayed:

- The volume label and serial number of the disk.
- One directory or file name per line, including the file name extension.
- The file size in bytes.
- The date and time the file was last modified.
- The total number of files listed and their cumulative size.
- The amount of free space (in bytes) remaining on the disk.

Type

DOS, Internal, Network

Syntax

dir [*drive*:][*path*][*filename*][/**p**][/**w**][/**a**[[:]*attributes*]][/**o**[[:]*sortorder*]][/**s**] [/**b**][/**l**]

Parameters

[***drive***:][***path***]

Specifies the drive and directory for which you want to see a listing.

[***filename***]

Specifies a particular file or group of files for which you want to see a listing.

Switches

/p Displays one screen of the listing at a time. To see the next screen, press any key.

/w Displays the listing in wide format, with as many as five file names or directory names on each line.

/**a**[[**:**]*attributes*]

Displays only the names of those directories and files with the attributes you specify. If you omit this switch, the names of all files except hidden and system files are displayed. If you use this switch without specifying *attributes*, the names of all files, including hidden and system files are displayed. The following list describes each of the values you can use for *attributes*. The colon (**:**) is optional. Use any combination of these values, and do not separate the values with spaces.

h	***Hidden files***
–h	***Files that are not hidden***
s	***System files***
–s	***Files other than system files***
d	***Directories***
–d	***Files only (not directories)***
a	***Files ready for archiving (backup)***
–a	***Files that have not changed since the last backup***
r	***Read-only files***
–r	***Files that are not read-only***

/**o[:]***sortorder*

Controls the order in which directory names and file names are sorted and displayed. If you omit this switch, the names are displayed in the order in which they occur in the directory. If you use this switch without specifying *sortorder*, the directory names followed by the file names are displayed, sorted in alphabetic order. The colon (**:**) is optional. The following list describes each of the values you can use for *sortorder*. Use any combination of the values, and do not separate these values with spaces.

n	In alphabetic order by name
–n	In reverse alphabetic order by name (Z through A)

e	In alphabetic order by extension
–e	In reverse alphabetic order by extension (Z through A)
d	By date and time, earliest first
–d	By date and time, latest first
s	By size, smallest first
–s	By size, largest first
g	With directories grouped before files
–g	With directories grouped after files

/s Lists every occurrence, in the specified directory and all subdirectories, of the specified file name.

/b Lists each directory name or file name, one per line (including the file name extension). This switch displays no heading information and no summary. The **/b** switch overrides the **/w** switch.

/l Displays unsorted directory names and file names in lowercase. This switch does not convert extended characters to lowercase.

Notes

Using wildcards with DIR

You can use wildcards (**?** and *****) to display a listing of a subset of files and subdirectories. For an example illustrating the use of a wildcard, see the following "Examples" section.

Specifying file display attributes

If you specify the **/a** switch with more than one value in *attributes*, the names of only those files with all the specified attributes are displayed. For example, if you specify the **/a** switch with the *r* and *–h* values for *attributes* by using either **/a**:r*–h* or **/a***r–h*, only the names of read-only files that are not hidden are displayed.

Specifying file name sorting

If you specify more than one *sortorder* value, the file names are sorted by the first criterion first, then by the second criterion, and so on. For example, if you specify the **/o** switch with the *e* and *–s* values for *sortorder* by using either **/o**:*e–s* or **/o***e–s*,, the names of the directories and files are sorted by extension, with the largest first. The alphabetic sorting by extension causes file names with no extensions to appear first, then directory names, then file names with extensions.

Setting date and time formats

The date and time formats used by the DIR command depend on the COUNTRY setting you use in your CONFIG.SYS file. If you do not use the COUNTRY command, the formats are those for the United States.

Using redirection symbols and pipes

When you use a redirection symbol (>) to send the output from a DIR command to a file or a pipe (|) to send the output to another command, use the **/a**:-*d* and **/b** switches to list only the file names. You can use the *filename* parameter with the **/b** and **/s** switches to specify that the current directory be searched along with its subdirectories for all file names that match the *filename* parameter.

Before using a pipe for redirection, you should set the *temp* environment variable in your AUTOEXEC.BAT file.

Presetting DIR parameters and switches

You can preset the DIR command parameters and switches by including the SET command with the *dircmd:* environment variable in your AUTOEXEC.BAT file. You can use any valid combination of DIR command parameters and switches with the SET *dircmd* command, including the location and name of a file.

For example, to use the *dircmd* environment variable to set the wide display format (**/w**) as the default format, include the following command in your AUTOEXEC.BAT file:

```
set dircmd=/w
```

For a single use of the DIR command, you can override a switch set by using the *dircmd* environment variable. To do so, you use the same switch on the DIR command line, but you must also precede the switch letter with a minus sign, as the following example shows:

```
dir /-w
```

You can change the *dircmd* default settings by typing the SET command at the command prompt with a new parameter or switch after the equal sign (=). The new default settings are effective for all subsequent DIR commands until you use SET *dircmd* again on the command line or until you restart DOS.

To clear all default settings, type the following command:

```
set dircmd=
```

You can view the current settings of the *dircmd* environment variable by typing the following command:

```
set
```

DOS displays a list of environment variables and their settings. For more information about setting environment variables, see the command “SET” on page 206.

Examples

Suppose you want to display one directory listing after another, until a listing for every directory on the disk in the current drive has been displayed. Suppose also that you want to alphabetize each directory listing, display it in wide format, and pause after each screen. To specify such a display, be sure the root directory is the current directory and then type the following command:

```
dir /s/w/o/p
```

The name of the root directory, the names of the subdirectories of the root directory, and the names of the files in the root directory (including extensions) are displayed. Then the subdirectory names and file names in each subdirectory in the directory tree are listed.

To alter the preceding example so that the file names and extensions are displayed but the directory names are omitted, type the following command:

```
dir /s/w/o/p/a:-d
```

To print a directory listing, type the redirection symbol and *prn* after any form of the DIR command, as the following example shows:

```
dir > prn
```

When you specify *prn* on the DIR command line, the directory listing is sent to the printer attached to the LPT1 port. If your printer is attached to a different port, you must replace *prn* with the name of the correct port.

You can also redirect output of the DIR command to a file by replacing *prn* with a file name. A path is also accepted on the command line. For example, to direct the output from the DIR command to the file DIR.DOC in the RECORDS directory, type the following command:

```
dir > \records\dir.doc
```

If DIR.DOC does not exist, DOS creates it, unless the directory RECORDS also does not exist. In that case, DOS displays the following message:

```
File creation error
```

To display a list of all the file names with the .TXT extension in all directories on drive C, type the following command:

```
dir c:\*.txt /w/o/s/p
```

DIR displays, in wide format, An alphabetized list of the matching file names in each directory is displayed in wide format. Each time the screen fills it pauses, until you press a key to continue.

Related Information

For information about displaying the directory structure of a path or disk, see the command "TREE" on page 229.

DISKCOMP

Compares the contents of two diskettes.

This command performs a track-by-track comparison. DISKCOMP determines the number of sides and sectors per track to compare based on the format of the first disk you specify.

Type

DOS, External

Syntax

diskcomp [*drive1*: [*drive2*:]] [/**1**] [/**8**]

Parameters

drive1: Specifies the drive containing one of the diskettes.

drive2: Specifies the drive containing the other diskette.

Switches

/**1** Compares only the first side of the diskettes, even if the diskettes are double-sided and the drives can read double-sided diskettes.

/**8** Compares only the first 8 sectors per track, even if the diskettes contain 9 or 15 sectors per track.

Notes

Invalid drive for DISKCOMP

The DISKCOMP command works only with diskettes. You cannot use the DISKCOMP command with a hard disk. If you specify a hard disk drive for *drive1* or *drive2*, the following error message is displayed:

```
Invalid drive specification
Specified drive does not exist
or is non-removable
```

DISKCOMP messages

If all tracks on the two diskettes being compared are the same, the following message is displayed:

```
Compare OK
```

If the tracks are not the same, the following message is displayed:

```
Compare error on
side 1, track 2
```

When the comparison is completed, the following message is displayed:

```
Compare another diskette (Y/N)?
```

If you press **Y**, you are prompted to insert diskettes for the next comparison. If you press **N**, the comparison is stopped.

The volume number of a diskette is ignored when a comparison is made.

Omitting drive parameters

If you omit the *drive2* parameter, the current drive for *drive2* is used. If you omit both drive parameters, the current drive is used for both. If the current drive is the same as *drive1*, you are prompted to swap diskettes as necessary.

Using one drive for the comparison

If you specify the same diskette drive for *drive1* and *drive2*, a comparison is done by using one drive. You are prompted to insert diskettes as necessary. You might have to swap diskettes more than once, depending on the capacity of the diskettes and the amount of available memory.

Comparing different types of diskettes

You cannot use the DISKCOMP command to compare a single-sided diskette with a double-sided diskette, nor a high-density diskette with a double-density diskette. If the diskette in *drive1* is not of the same type as the diskette in *drive2*, the following message is displayed:

```
Drive types or diskette types not compatible
```

Using DISKCOMP with networks and redirected drives

You cannot use the DISKCOMP command a network drive or on a drive created or affected by the ASSIGN. JOIN, or SUBST commands. If you attempt to use the DISKCOMP command with a drive of any of these types, an error message is displayed.

Comparing an original diskette with a copy

When you use the DISKCOMP command with a diskette that you made with the COPY command, a message similar to the following might be displayed:

```
Compare error on
side 0, track 0
```

This type of error can occur even if the files on the diskettes are identical. Although the COPY command duplicates information, it does not necessarily place it in the same location on the destination diskette. For more information about comparing individual files on two diskettes, see “FC” on page 105.

DISKCOMP exit codes

The following list shows each exit code and a brief description of its meaning:

0 The diskettes are the same.

1 Differences were found.

2 The user pressed **Ctrl+C** to stop the process.

3 A hard error occurred.

4 An initialization error occurred.

You can use the *errorlevel* parameter on the **IF** command line in a batch program to process exit codes returned by using the DISKCOMP command. For an example of a batch program that processes exit codes, see the following "Examples" section.

Examples

If your system has only one diskette drive, drive A, and you want to compare two diskettes, type the following command:

```
diskcomp a: a:
```

You are prompted to insert each diskette, as required.

Following is an example of a batch program that uses the *errorlevel* parameter on the **IF** command line to process a DISKCOMP exit code:

```
rem CHECKOUT.BAT compares the diskettes in drive A and B
echo off
diskcomp a: b:
if errorlevel 4 goto ini_error
if errorlevel 3 goto hard_error
if errorlevel 2 goto break
if errorlevel 1 goto no_compare
if errorlevel 0 goto compare_ok
:ini_error
echo ERROR: Insufficient memory or command invalid
goto exit
:hard_error
echo ERROR: An unrecoverable error occurred
goto exit
:break
echo You just pressed Ctrl+C to stop the comparison
goto exit
:no_compare
echo Diskettes are not the same
goto exit
:compare_ok
echo The comparison was successful; the diskettes are the same
goto exit
:exit
```

Related Information

For information about comparing two files, see the commands "COMP" on page 44 and "FC" on page 105.

For more information about working with batch programs see the *PC DOS User's Guide*.

DISKCOPY

Copies the contents of the diskette in the source drive to a formatted or unformatted diskette in the destination drive. Using the DISKCOPY command destroys the existing contents of the destination diskette as it copies the new information to it.

This command determines the number of sides to copy based on the source drive and diskette.

Type

DOS, External

Syntax

DISKCOPY [*drive1*: [*drive2*:]] [/**1**] [/**v**]

Parameters

drive1: Specifies the drive containing the source diskette.

drive2: Specifies the drive containing the destination diskette.

Switches

/1 Copies only the first side of a diskette.

/v Verifies that the information is copied correctly. Use of this switch slows the copying process.

Notes

Invalid drive for DISKCOPY

The DISKCOPY command works only with removable disks, such as diskettes. You cannot use the DISKCOPY command with a hard disk. If you specify a hard disk drive for *drive1* or *drive2*, the following error message is displayed:

```
Invalid drive specification
Specified drive does not exist
or is non-removable
```

DISKCOPY messages

The DISKCOPY command prompts you to insert the source and destination diskettes and waits for you to press any key before continuing.

After copying, the following message is displayed:

```
Copy another diskette (Y/N)?
```

If you press **Y**, you are prompted to insert source and destination diskettes for the next copy operation. To stop the DISKCOPY process, press **N**.

If you are copying to an unformatted diskette in *drive2*, the diskette is formatted with the same number of sides and sectors per track as are on the diskette in *drive1*. The following message is displayed while the diskette is formatted and the files copied:

```
Formatting while copying
```

If the capacity of the source diskette is greater than that of the destination diskette and your computer can detect this difference, the following message is displayed:

```
TARGET media has lower capacity than SOURCE
Continue anyway (Y/N)?
```

If you press **Y**, an attempt is made to format the destination diskette and copy the files.

Diskette serial numbers

If the source diskette has a volume serial number, a new volume serial number is created for the destination diskette. This number is displayed when the copy operation is complete.

Omitting drive parameters

If you omit the *drive2* parameter, the current drive is used as the destination drive. If you omit both drive parameters, the current drive is used for both. If the current drive is the same as *drive1*, you are prompted to swap diskettes as necessary.

Using one drive for copying

If *drive1* and *drive2* are the same, you are prompted whenever you should switch disks. If you omit both drive parameters and the current drive is a diskette drive, you are prompted each time you should insert a diskette in the drive. If the diskettes contain more information than available memory can hold, all of the information cannot be read at once. The DISKCOPY process reads from the source diskette, writes to the destination diskette, and you are prompted to insert the source diskette again. This process continues until the entire diskette has been copied.

Avoiding diskette fragmentation

Because an exact copy of the source diskette is made on the destination diskette, any *fragmentation* on the source disk is transferred to the destination diskette. Fragmentation is the presence of small areas of unused diskette space between existing files on a diskette.

A fragmented source diskette can slow down the finding, reading, or writing of files. To avoid transferring fragmentation from one diskette to another, use either the COPY command or XCOPY command to copy your diskette. Because COPY and XCOPY copy files sequentially, the new diskette is not fragmented.

CAUTION:
You cannot use XCOPY to copy a startup disk.

DISKCOPY exit codes

The following list briefly describes the meaning of each DISKCOPY exit code (*errorlevel*):

0	The copy operation was successful.
1	A nonfatal read/write error occurred.
2	The user pressed **Ctrl+C** to stop the process.
3	A fatal hard error occurred.
4	An initialization error occurred.

You can use the *errorlevel* parameter on the **IF** command line in a batch program to process exit codes returned by using the DISKCOPY command. For an example of a batch program that processes exit codes, see command "DISKCOMP" on page 76.

Related Information

For information about copying one or more files, see the command "COPY" on page 47.

For information about copying directories and subdirectories, see the command "XCOPY" on page 240

DISPLAY.SYS

This is one of the installable device drivers provided with DOS. It supports code page switching for displays.

Related Information

For detailed information, see "DISPLAY.SYS" on page 254.

DOS

Specifies that DOS is to maintain a link to the upper memory area or is to load part of itself into the high memory area (HMA).

Type

CONFIG.SYS

Syntax

dos=*high*|*low*[,*umb*|,*noumb*]

dos=[*high*,|*low*,]*umb*|*noumb*

Parameters

umb*|*noumb

Specifies whether DOS should maintain a link between conventional memory and the upper memory area. The *umb* parameter provides this link. The *noumb* parameter disconnects this link. The default setting is *noumb*.

high|low

Specifies whether DOS should attempt to load a part of itself into the HMA. Use the *high* parameter to enable DOS to load itself into the HMA. Use the *low* parameter to keep all of DOS in conventional memory. The default setting is *low*.

Notes

Must install HIMEM.SYS for DOS=*high*

You must install the HIMEM.SYS device driver before you specify either DOS=*umb* or DOS=*high*.

Using the *umb* parameter

You must specify the DOS=*umb* command in order to load programs and device drivers into the upper memory area. Using the upper memory area frees more space in conventional memory for programs. In addition to using this command, you must install an upper-memory-block (UMB) provider.

Using the *high* parameter

If you specify the *high* parameter, DOS attempts to load part of itself into the HMA. Loading part of DOS into the HMA frees conventional memory for programs.

Combining parameters

You can include more than one parameter on a single DOS command line, using commas to separate them. For example, the following command lines are valid:

```
dos=umb,low
```

```
dos=high,umb
```

You can place the DOS command anywhere in your CONFIG.SYS file.

Related Information

For information about loading a device driver into the upper memory area, see the command "DEVICEHIGH" on page 70.

For information about loading a program into the upper memory area, see the command "LOADHIGH" on page 146.

DOSKEY

Starts the Doskey program, which recalls DOS commands, edits command lines, and creates macros.

The Doskey program is a terminate-and-stay-resident program. You can use this program to customize and automate DOS command lines. When installed, the Doskey program occupies about 3 kilobytes of resident memory.

Type

DOS, External, Network

Syntax

doskey [**/reinstall**] [**/bufsize=***size*] [**/macros**] [**/history**] [**/insert**|**/overstrike**] [*macroname***=**[*text*]]

To start the Doskey program and use the default settings, use the following syntax:

doskey

Parameters

***macroname*=[*text*]**
Creates a macro that carries out one or more DOS commands (a Doskey macro). *Macroname* specifies the name you want to assign to the macro. *Text* specifies the commands you want to record.

Switches

/reinstall
Installs a new copy of the Doskey program, even if one is already installed. In the latter case, the **/reinstall** switch also clears the buffer.

/bufsize=*size*
Specifies the size of the buffer in which Doskey stores commands and Doskey macros. The default size is 512 bytes. The minimum buffer size is 256 bytes.

/macros
Displays a list of all DOSKEY macros. You can use a redirection symbol (>) with the ***/macros*** switch to redirect the list to a file. You can abbreviate the ***/macros*** switch as ***/m***.

/history
Displays a list of all commands stored in memory. You can use a redirection symbol (>) with the /**history** switch to redirect the list to a file. You can abbreviate the /**history** switch as /**h**.

/insert|/overstrike
Specifies whether new text you type is to replace old text. If you use the /**insert** switch, new text that you type on a line is inserted into old text (as if you had pressed the **Ins** key). If you use the /**overstrike** switch, new text replaces old text. The default setting is /**overstrike**.

Notes

Recalling a command

To recall a command, you can use any of the following keys after loading Doskey into memory:

Up Arrow
Recalls the DOS command you used before the one displayed.

Down Arrow
Recalls the DOS command you used after the one displayed.

PgUp Recalls the oldest DOS command you used in the current session.

PgDn Recalls the most recent DOS command you used.

Editing the command line

With the Doskey program, you can edit the current command line. The following list describes the DOSKEY editing keys and their functions:

Left Arrow
Moves the cursor back one character.

Right Arrow
Moves the cursor forward one character.

Ctrl+Left Arrow
Moves the cursor back one word.

Ctrl+Right Arrow
Moves the cursor forward one word.

Home Moves the cursor to the beginning of the line.

End Moves the cursor to the end of the line.

Esc Clears the command from the display.

F1 Copies one character from the *template* to the DOS command line. (The template is a memory buffer that holds the last command you typed.)

F2 Searches forward in the template for the next key you type after pressing **F2**. Doskey inserts the text from the template up to but not including the character you specify.

F3 Copies the remainder of the template to the command line. Doskey begins copying characters from the position in the template that corresponds to the position indicated by the cursor on the command line.

F4 Deletes characters, beginning with the first character in the template, up to a character you specify. To use this editing key, you press **F4** and type a character. Doskey deletes up to, but not including, that character.

F5 Copies the current command into the template and clears the command line.

F6 Places an end-of-file character (**Ctrl+Z**) at the end of the current command line.

F7 Displays all commands stored in memory, with their associated numbers. Doskey assigns these numbers sequentially, beginning with 1 for the first (oldest) command stored in memory.

Alt+F7 Deletes all commands stored in memory.

F8 Searches memory for a command that you want Doskey to display. To use this editing key, type the first character, or the first few characters, of the command you want Doskey to search for and then press **F8**. Doskey displays the most recent command that begins with the text you typed. Press **F8** repeatedly to cycle through all the commands that start with the characters you specified.

F9 Prompts you for a command number and displays the command associated with the number you specify. To display all the numbers and their associated commands, press **F7**.

Alt+F10
Deletes all macro definitions.

Specifying a default insert mode

If you press the **Ins** key, you can type text on the DOSKEY command line in the middle of old text without replacing the old text. However, once you press **Enter**, Doskey returns your keyboard to replace mode. You must press **Ins** again to return to insert mode.

The **/insert** switch puts your keyboard in insert mode each time you press **Enter**. Your keyboard effectively remains in insert mode until you use the **/overstrike** switch. You can temporarily return to replace mode by pressing the **Ins** key; but once you press **Enter**, Doskey returns your keyboard to insert mode.

The cursor changes shape when you use the **Ins** key to change from one mode to the other.

Creating a macro

You can use the Doskey program to create macros that carry out one or more DOS commands.

You can use the following special characters to control command operations when defining a macro:

$G or **$g**
Redirects output. Use either of these special characters to send output to a device or a file instead of to the screen. This character is equivalent to the redirection symbol for output (>).

GG or **gg**
Appends output to the end of a file. Use either of these special double characters to append output to an existing file rather than replace the data in the file. These double characters are equivalent to the "append" redirection symbol for output (>>).

$L or **$l**
Redirects input. Use either of these special characters to read input from a device or a file instead of from the keyboard. This character is equivalent to the redirection symbol for input (<).

$B or **$b**
Sends macro output to a command. Using one of these special characters is equivalent to using the pipe (|) on a command line.

$T or **$t**
Separates commands. Use either of these special characters to separate commands when you are creating macros or typing commands on the DOSKEY command line.

**$$** Specifies the dollar-sign character ($).

$1 through **$9**
Represent any command-line information you want to specify when you run the macro. The special characters *$1* through *$9* are batch parameters, which make it possible for you to use different data on the command line each time you run the macro. The *$1* character in a DOSKEY command is similar to the *%1* character in a batch program.

$* Represents *all* the command-line information you want to specify when you type the macro name. The special character *$** is a replaceable parameter that is similar to the batch parameters *$1* through *$9*, with one important difference. Here, *everything* you type on the command line after the macro name is substituted for the *$** in the macro.

For example, to create a macro that performs a quick and unconditional format of a disk, type the following command:

```
doskey qf=format $1 /q /u
```

For information about quick and unconditional formatting, see the command "FORMAT" on page 115.

You can use the DOSKEY command in a batch program to create a macro.

Running a macro

To run a macro, type the macro name starting at the first position on the command line. If the macro was defined with *$** or any of the batch parameters *$1* through *$9*, use a space to separate parameters.

You could run the QF macro created in the previous example to format a disk in drive A quickly and unconditionally. To do so, you would type the following command:

```
qf a:
```

You cannot run a macro from a batch program.

Creating a macro with the same name as a DOS command

You might want to create a macro that has the same name as a DOS command. This can be useful, for example, if you always use a certain command with specific switches. To specify whether you want to run the macro or the DOS command, follow these guidelines:

- To run the macro, begin typing the macro name immediately after the command prompt, with no space between the prompt and the command name.
- To carry out the command, insert one or more spaces between the command prompt and the command name.

Deleting a macro

To delete a macro, type the following command:

```
doskey macroname=
```

Examples

The **/macros** and **/history** switches are useful for creating batch programs to save macros and commands. For example, to create a batch program named MACINIT.BAT that includes all DOSKEY macros, type the following command:

```
doskey /macros > macinit.bat
```

To use the MACINIT.BAT file, edit it to include the DOSKEY command at the beginning of each macro line.

To create a batch program named TMP.BAT that contains recently used commands, type the following command:

```
doskey /history > tmp.bat
```

To define a macro with multiple commands, use *$t* to separate commands, as follows:

```
doskey tx=cd\temp$tdir/w $*
```

In the preceding example, the TX macro changes the current directory to TEMP and then displays a directory listing, using the wide display format. You can use *$** at the end of the macro to append other switches to the DIR command when you run the TX macro.

The following macro uses a batch parameter for a new directory name. The macro first creates a new directory and then changes to it from the current directory.

```
doskey mc=md $1$tcd $1
```

To use the preceding macro to create and change to a directory named BOOKS, you type the following:

```
mc books
```

To create a macro that uses batch parameters for moving a file or group of files, type the following command:

```
doskey mv=copy $1 $2 $t del $1
```

DOSSHELL

Starts DOS Shell, a graphical interface to DOS.

Type

DOS, External, Network

Syntax

To start DOS Shell in text mode, use the following syntax:

dosshell [/**t**[:*res*[*n*]]] [/**b**]

To start DOS Shell in graphics mode, use the following syntax:

dosshell [/**g**[:*res*[*n*]]] [/**b**]

Parameters

res Specifies a screen-resolution category. Valid values are *l*, *m*, and *h* to specify low, medium, and high resolution, respectively. The default value of *res* depends on your hardware.

n Specifies a screen resolution when there is more than one choice within a category. For information about the valid values for this parameter, see the following "Note" section. The default value of *n* depends on your hardware.

Switches

/t Starts DOS Shell in text mode.

/b Starts DOS Shell using a black-and-white color scheme.

/g Starts DOS Shell in graphics mode.

Note

Once you have started DOS Shell, you can adjust the screen resolution by using the DISPLAY command on the Options menu. A dialog box displays the mode (text or graphics), the number of lines, the resolution category, and the specific number within each category for all possible screen-resolution modes available for your hardware.

Examples

To start DOS Shell in graphics mode, type the following command:

```
dosshell /g
```

DRIVER.SYS

This is one of the installable device drivers provided with DOS. It creates a logical drive that you can use to refer to a physical disk driver and specifies parameters for a drive not supported by your computer's ROM BIOS.

Related Information

For detailed information, see "DRIVER.SYS" on page 255.

DRIVPARM

Defines parameters for block devices when you start DOS.

The DRIVPARM command modifies the parameters of an existing physical drive. It does not create a new logical drive. The settings specified in the DRIVPARM command override the driver definitions for any previous block device.

Type

CONFIG.SYS

Syntax

drivparm=/d:*number* [**/c**][**/f**:*factor*][**/h**:*heads*][**/i**] [**/n**][**/s**:*sectors*][**/t**:*tracks*]

Switches

/d:*number*

Specifies the physical drive number. Values for *number* must be in the range 0 through 255 (for example, drive number 0 = drive A, 1 = drive B, 2 = drive C, and so on).

/c Specifies that the drive can detect whether the drive door is closed.

/f:*factor*

Specifies the drive type. The following list shows the valid values for *factor* and a brief description of each. The default value is 2.

0	160K/180K or 320K/360K
1	1.2MB
2	720K (3.5-inch diskette)
5	Hard disk
6	Tape
7	1.44MB (3.5-inch diskette)
8	Read/write optical disk
9	2.88MB (3.5-inch diskette)

/h:*heads*

Specifies the maximum number of heads. Values for *heads* must be in the range 1 through 99. The default value depends upon the value you specify for **/f:***factor.*

/i Specifies an electronically-compatible 3.5-inch diskette drive. (Electronically-compatible drives are installed on your computer and use your existing diskette-drive controller.) Use the **/i** switch if the ROM BIOS of your computer does not support 3.5-inch diskette drives.

/n Specifies a nonremovable block device.

/s:*sectors*

Specifies the number of sectors per track that the block device supports. Values for *sectors* must be in the range 1 through 99. The default value depends upon the value you specify for **/f:***factor.*

/t:*tracks*

Specifies the number of tracks per side that the block device supports. The default value depends upon the value you specify for **/f:***factor.*

Notes

Using the /i switch

Use the **/i** switch if your system does not support 3.5-inch diskette drives. (Some IBM AT*-compatible systems do not have a ROM BIOS that supports 3.5-inch diskette drives.)

Diskette drive change-line support

Change-line support means that a physical diskette drive can detect whether the drive door is open. Change-line support improves performance by letting DOS know when one diskette has been replaced by another. The **/c** switch enables DOS to make use of change-line support. To find out whether your diskette drive has change-line support, see your diskette-drive documentation.

Creating a logical drive

DRIVPARM modifies the parameters of an existing physical drive and does not create a new logical drive. For information about creating a new logical drive and associating it with a physical drive, see “Customizing Your System” in the *User's Guide*.

Examples

Suppose your system has an internal tape drive with one head on drive D that is configured at startup to write 20 tracks of 40 sectors per track. To reconfigure this tape drive to write 10 tracks of 99 sectors each, add the following command to your CONFIG.SYS file:

```
drivparm=/d:3 /f:6 /h:1 /s:99 /t:10
```

DRVLOCK

Locks or unlocks the specified drive or socket. When used to lock a drive, it is so the media can be secured. When used to lock a socket, it prevents PCMCIA PC Card removal.

Type

DOS

Syntax

drvlock [*drive*:|*socket*:] [**/on** |**/off**]

Parameters

***drive*:** Specifies a drive to lock or unlock. If no drive or socket is specified, DRVLOCK defaults to the current drive.

***socket*:**
Specifies a PCMCIA socket to be locked or unlocked. If no drive or socket is specified, DRVLOCK defaults to the current drive.

Switches

/on Turns lock on.

/off Turns lock off.

Notes

No switch is specified

If no switch is specified, DRVLOCK reports the lock or unlock status of the current drive.

Drive or socket is already locked

If the drive or socket is already locked, DRVLOCK does not attempt to lock it again. A message is displayed informing you that the drive or socket is locked.

E

Starts the DOS Editor, which is used to create and change ASCII text files.

The DOS Editor is a full-screen editor that enables you to create, edit, save, and prinı ASCII text files.

Type

DOS, External, Network

Syntax

e [=][*filename*][**/Q**]

Parameters

filename Specifies the name of the file to be edited. Multiple file names can be specified. Wildcards are allowed.

Switches

= Use the same path as last specified.

/Q Suppresses the display of "Loading..." message.

Examples

The equal sign (=) can be used to specify more than one file on the command line when invoking the editor to save keystrokes if both files have the same path. The following example shows the use of the equal sign (=).

```
e \c\src\prog.c =prog.h
```

For additional information about using the DOS E editor, see the *PC DOS User's Guide*.

ECHO

Turns the command-echoing feature on or off, or displays a message.

When you run a batch program, DOS typically displays (echoes) the commands of the batch program on the screen. You can turn this feature on or off by using the ECHO command.

Type

Batch, Internal

Syntax

echo [**on**|**off**]

To use the ECHO command to display a message, use the following syntax:

echo [*message*]

Parameters

message
Specifies text you want DOS to display on the screen.

Switches

on|off Specifies whether to turn the command-echoing feature on or off. To display the current ECHO setting, use the ECHO command without a parameter.

Notes

Using a message with the ECHO command

The ECHO *message* command is useful when ECHO is off. To display a message that is several lines long without displaying other commands, you can include several ECHO *message* commands after the ECHO **off** command in your batch program.

Hiding the command prompt

If you use the ECHO **off** command on the command line, the command prompt does not appear on your screen. To re-display the command prompt, type **echo on**.

Preventing DOS from echoing a line

You can insert an at sign (@) in front of a command in a batch program to prevent DOS from echoing that line.

Echoing a blank line

To echo a blank line on the screen, you can type ECHO and then a period (**echo.**). There must be no intervening space.

Displaying pipes and redirection characters

You cannot display a pipe (|) or redirection character (> or <) by using the ECHO command.

Examples

The following example shows a batch program that includes a three-line message preceded and followed by a blank line:

```
echo off
   echo.
   echo This batch program
   echo formats and checks
   echo new disks
   echo.
```

If you want to turn ECHO off and you do not want to echo the ECHO command itself, include an at sign (@) before the command, as follows:

```
   @echo off
```

You can use the IF and ECHO commands on the same command line, as follows:

```
   if exist *.rpt echo The report has arrived.
```

Related Information

For information about suspending the execution of a batch program, see the command "PAUSE" on page 176.

For information about using the ECHO command in batch programs, see the *PC DOS User's Guide*.

EDLIN

The EDLIN command plus a filename starts the EDLIN editor, a line-oriented text editor with which you can create and change ASCII files.

The EDLIN editor numbers each line of the text file that is located in memory. You can use this editor to insert, modify, copy, move, and delete lines of the file. If you want to use a full-screen editor, use the E command.

Type

DOS, External, Network

Syntax

edlin [*drive*:][*path*]*filename* [/**b**]

Parameters

[***drive***:][*path*] *filename*

Specifies the location and name of an ASCII file on a disk. If the file exists, it is opened. If the file does not exist, a file is created in memory. The specified location and file name is used by the editor to create the file on a disk when you use the EDLIN E command.

Switches

/b Specifies that EDLIN is to ignore the end-of-file character (**Ctrl+Z**).

EDLIN commands

The following information is provided as a quick reference for the EDLIN commands.

Syntax	**Command**
?	? (EDLIN COMMANDS): Displays a list of the EDLIN commands after specifying the filename you want to edit.
Edit line	line#
[#lines]A	A (APPEND LINES): Adds a specified number of lines from disk to the file being edited in storage when insufficient memory prohibits loading the entire file.
[startline],[endline],toline[,times]C	C (COPY LINES): Copies the lines in the specified range to the line number specified by the *toline* parameter. *times* specifies the number of times you want to repeat the text.
[startline][endline]D	D (DELETE LINES): Deletes the lines in the specified range, making the line following the deleted range the current line.
E	E (END EDIT): Ends EDLIN and saves the edited file.
[line]i	I (INSERT LINE): Inserts the specified line of text before the line number you specify in the edited file in memory.
[startline][,endline]L	L (LIST LINES): Displays the specified range of lines, leaving the current line unchanged.
[startline],[endline],tolineM	M (MOVE LINES): Moves the lines in the specified range to the line number specified by the *toline* parameter.
[startline][,endline]P	P (PAGE): Lists the specified range of lines, changing the current line.

Syntax	Command
Q	Q (QUIT EDIT): Quits the file you are editing without saving any changes that you have entered.
[startline][,endline][?]R [oldtext][**Ctrl+Z**newtext]	R (REPLACE TEXT): Replaces all occurrences of the old text in the specified range of lines, with the new text. The ? requests a prompt "OK?" after each display of a changed line.
[startline][,endline][?]Stext	S (SEARCH TEXT): Searches the specified range of lines to find the specified text.
[toline]T[drive:][path]filename	T (TRANSFER LINES): Transfers (merges)the contents of the specified file (*filename*) ahead of the line in the file you are currently editing.
[#lines]W	W (WRITE LINES): Writes the specified number of lines to disk from the lines being edited in storage.

Notes

Maximum line length

EDLIN accepts a maximum of 253 characters per line.

Meaning of the asterisk character in EDLIN

The asterisk (*) is used for two purposes in EDLIN. When an asterisk appears as the only character on the display line, it is the EDLIN prompt after which you type EDLIN commands. When an asterisk appears after a line number on the display line, it indicates that the line is the current line (where the cursor is located).

Meaning of a page of text

A *page* of text is one full screen of information. With a 25-line screen mode, EDLIN displays 24 lines of text per page. The number of lines per page depends on the screen mode you are using.

Starting and stopping insert mode

To insert lines into the file in memory, use the EDLIN I (INSERT) command. Once you have finished inserting lines, press **Enter** and then **Ctrl+C** to stop the insert mode.

EGA.SYS

This is one of the installable device drivers provided with DOS. It saves and restores the display when a graphics program is used with an EGA monitor.

Related Information

For detailed information, see "EGA.SYS" on page 257.

EJECT

Ejects the media from a drive.

If no drive is specified, the media is ejected from the current drive.

Type

DOS

Syntax

eject [*drive*:]

Parameters

[*drive*:]
Specifies the drive letter of the drive you want to eject media from.

Notes

The media cannot be ejected from the drive specified when the drive is locked. See command "DRVLOCK" on page 92.

EMM386

Enables or disables EMM386 expanded-memory support on a computer with an 80386 or higher processor.

Type

DOS, External

Syntax

emm386 [on|off|auto][w=on|w=off]

To display the current status of EMM386 expanded-memory support, use the following syntax:

emm386

Parameters

on|off|auto

Activates the EMM386.EXE device driver (if set to *on*), suspends the EMM386.EXE device driver (if set to *off*), or places the EMM386.EXE device driver in auto mode (if set to *auto*). Auto mode enables expanded memory support only when a program calls for it. The default value is **on**.

w=on|w=off

Enables or disables Weitek coprocessor support. The default value is **w=off**.

Notes

Installing the EMM386.EXE device driver

You must use the DEVICE command to install the EMM386.EXE device driver before you can use the EMM386 command. To use the EMM386.EXE device driver and the EMM386 command, your computer must have an 80386 or higher processor. If you try to use the EMM386 command on a computer that does not have an 80386 or higher processor, DOS displays the following message:

```
EMM386 driver not installed
```

Reactivating EMM386 expanded-memory support

If EMM386.EXE was loaded when DOS started but is not currently in use, the **on** parameter reactivates expanded-memory support.

Suspending EMM386 expanded-memory support

If EMM386 expanded-memory support is currently active, handle 0 is the only handle allocated, and EMM386.EXE is not providing access to the upper memory area, then the **off** parameter suspends EMM386 expanded-memory support. When EMM386 expanded-memory support is off, the Lotus**/Intel**/Microsoft**/AST** Expanded Memory Specification (LIM EMS) header is changed so that programs cannot use expanded memory.

Enabling and disabling Weitek coprocessor support

If the **w=on** parameter is specified and the **off** parameter (different from the **w=off** parameter) is not, EMM386 enables Weitek coprocessor support. The high memory area (HMA) must be available to enable Weitek coprocessor support.

If you specify the **w=on** or **w=off** parameter and no Weitek coprocessor is installed in your computer system, DOS displays the following error message:

```
Weitek Coprocessor not installed
```

EMM386.EXE

This is one of the installable device drivers provided with DOS. It simulates expanded memory and provides access to the upper memory area on a computer with at least an 80386 processor and extended memory.

Related Information

For a detailed explanation, see "EMM386.EXE" on page 258.

ERASE

Deletes one or more files. It works in the same way with the same syntax and options as the DEL command.

For more details see the command "DEL (ERASE)" on page 66.

EXE2BIN

Converts .EXE (executable) files to binary format.

EXE2BIN is included with DOS as a courtesy to software developers. It is not useful for general users.

Type

DOS, External, Network

Syntax

exe2bin [*drive1*:][*path1*]*input-file* [[*drive2*:][*path2*]*output-file*]

Parameters

[*drive1*:][*path1*]*input-file*
Specifies the location and name of the input file.

[*drive2*:][*path2*]*output-file*
Specifies the location and name of the output file.

Notes

Restrictions on using EXE2BIN

The following restrictions apply when you use the EXE2BIN command:

- The input file must be in valid .EXE format produced by the linker and must not be packed.
- The resident, or actual, code and data portions of the file combined must be less than 64K.
- There must be no STACK segment.

Default values for parameters

EXE2BIN takes specific actions, depending upon the values you use for the *input-file* and *output-file* parameters.

- The default file name extension for the file name you specify for *input-file* is .EXE. EXE2BIN converts the input .EXE file to an output file in .BIN format (a memory image of the program) and uses the location and file name you specify for [*drive2*:][*path2*]*output-file* to store that output file.
- If you do not specify *drive2* or *path2*, EXE2BIN writes the output file to the current drive and directory.
- If you do not specify an output file name, EXE2BIN uses the input file name.
- The default extension for the file name specified for the *output-file* parameter is .BIN.

Types of conversion available with EXE2BIN

Two types of conversion are possible, depending upon whether the initial CS:IP (Code Segment :Instruction Pointer) is specified in the .EXE file. The following list presents the two types:

- If the CS :IP is not specified in the .EXE file, EXE2BIN performs a pure binary conversion. If segment fixes are necessary (that is, if the program contains instructions requiring segment relocation), EXE2BIN prompts you for the fixup value. This value is the absolute segment at which the program is to be loaded. The resulting program is usable only when loaded at the absolute memory address specified by your program. The command interpreter cannot load the program.
- If the CS :IP is specified as *hex*0000:100, the file runs as a .COM file with the instruction pointer set at *100* by the assembler statement **org**. Include the .COM extension in the *output-file* parameter. No segment fixes are allowed, because .COM files must be segment-relocatable. The command interpreter can then load and run the program in the same way as it loads and runs the .COM programs supplied on your DOS disk.

EXIT

Quits the COMMAND.COM program (the command interpreter) and returns to the program that started COMMAND.COM, if one exists.

Type

DOS, Internal, Network

Syntax

exit

Notes

Using exit with the command interpreter

When you use the DOS COMMAND command to start a new command interpreter, you can use the EXIT command to return to the old command interpreter. Also, while running some programs, you can run the DOS command interpreter and then use the EXIT command to return to your program. For more information about command interpreters, see command "COMMAND" on page 40.

Using EXIT when the program is loaded as permanent

If you start the COMMAND.COM program with the **/p** (permanent) switch, the EXIT command has no effect.

EXPAND

Expands a compressed DOS file. You can use this command to retrieve one or more files from the installation diskettes that accompany DOS. These files are not usable unless you expand them.

Type

DOS, External, Network

Syntax

expand [*drive*:][*path*]*filename* [[*drive*:][*path*]*filename*[...]] *destination*

Parameters

[*drive*:][*path*]*filename*
Specifies the location and name of a compressed file or files to be expanded. You cannot use wildcards (* and ?).

destination
Specifies either or both the location and name of the expanded file or files. *Destination* can consist of a drive letter and colon, a directory name, a file name, or a combination. However, you cannot specify a file name for *destination* unless you also specify a single compressed file for file name.

Notes

Most of the files on the installation diskettes provided with DOS are compressed. Each of these compressed files has a file extension that ends with an underscore character (_). When you installed DOS, you ran the Setup program, which expanded these files before copying them to your system. However, if you need to retrieve just one or a few files from the original diskettes, you can use the EXPAND command.

Examples

Suppose you accidentally delete the SORT.EXE file from your DOS directory on drive C.

You can copy the compressed file SORT.EX_ from the DOS. installation diskettes to your hard disk. First, you need to find out which diskette contains SORT.EX_. To do this, use the DIR command on your DOS diskettes to determine the one that contains the SORT.EX_ file. After you find the appropriate diskette, use the following command to copy and expand the file:

```
expand a:\sort.ex_ c:\dos\sort.exe
```

FASTOPEN

Starts the Fastopen program, which decreases the amount of time needed to open frequently used files.

Fastopen tracks the location of files on a hard disk and stores the information in memory for fast access.

Type

DOS, External

Syntax

fastopen *drive*:[[=]*n*] [*drive*:[[=]*n*][...]][/**x**]

In your CONFIG.SYS file, use the following syntax:

install=[[*dos-drive*:]*dos-path*]**fastopen.exe** *drive*:[[=]*n*] [*drive*:[[=] *n*][...]] [/**x**]

Parameters

drive: Specifies a hard disk drive for which you want Fastopen to track the opening of files.

n Specifies the number of files Fastopen can work with at the same time. Valid values for *n* are in the range 10 through 999. The default value is 48.

[*dos-drive*:]*dos-path*
Specifies the location of FASTOPEN.EXE.

Switches

/x Creates the name cache in expanded memory instead of in conventional memory. The *name cache* is an area of memory in which DOS stores (caches) the locations and names of the files that you open. This cache conforms to version 4.0 of the Lotus/Intel/Microsoft Expanded Memory Specification (LIM EMS).

Notes

How Fastopen tracks information

Every time you open a file, Fastopen records its name and location in the name cache. If you later reopen a file recorded by Fastopen, the access time is greatly reduced.

Limits on using Fastopen

Fastopen works only on hard disks and does not work over a network. You can use Fastopen with as many as 24 hard-disk partitions at one time. For each partition, Fastopen can track the number of files specified by the *n* parameter.

You cannot run more than one copy of Fastopen at the same time. If you want to change the Fastopen settings, you must restart DOS.

You should not use the FASTOPEN command from DOS Shell, because doing so can lock up your machine.

Memory requirements for Fastopen

Fastopen requires approximately 48 bytes of memory for each file that it tracks.

Adding the FASTOPEN command to the CONFIG.SYS file

You can add a FASTOPEN command to your CONFIG.SYS file by using the INSTALL command. Use this technique when you do not want to start Fastopen from the DOS command line or from your AUTOEXEC.BAT file.

Cannot run disk-compaction program

To avoid losing data, do not run a disk-compaction program while FASTOPEN.EXE is loaded.

Examples

If you want DOS to track the location of as many as 100 files on drive C, add the following line to your CONFIG.SYS file:

```
install=c:\dos\fastopen.exe c:=100
```

FC

Compares two files and displays the differences between them.

Type

DOS, External, Network

Syntax

To make an ASCII comparison, use the following syntax:

fc [**/a**][**/c**][**/l**][**/lb***n*] [**/n**][**/t**][**/w**] [*/nnnn*][*drive1*:][*path1*]*filename1* [*drive2*:][*path2*]*filename2*

To make a binary comparison, use the following syntax:

fc /b [*drive1*:][*path1*]*filename1* [*drive2*:][*path2*]*filename2*

Parameters

[*drive1*:][*path1*]*filename1*
Specifies the location and name of the first file you want to compare.

[*drive2*:][*path2*]*filename2*
Specifies the location and name of the second file you want to compare.

Switches

/a Abbreviates the output of an ASCII comparison. Instead of displaying all the lines that are different, FC displays only the first and last line for each set of differences.

/c Ignores the case of letters.

/l Compares the files in ASCII mode. FC compares the two files line by line and attempts to resynchronize the files after finding a mismatch. This is the default mode for comparing files that do not have extensions of .EXE, .COM, .SYS, .OBJ, .LIB, or .BIN.

/lb*n* Sets the number of lines for the internal line buffer. The default length of the line buffer is 100 lines. If the files being compared have more than this number of consecutive differing lines, FC cancels the comparison.

/n Displays the line numbers during an ASCII comparison.

/t Does not expand tabs to spaces. The default behavior is to treat tabs as spaces, with stops at each eighth character position.

/w Compresses white space (tabs and spaces) during the comparison. If a line contains many consecutive spaces or tabs, the **/w** switch treats these characters as a single space. When used with the **/w** switch, FC ignores (and does not compare) white space at the beginning and end of a line.

/nnnn Specifies the number of consecutive lines that must match before FC considers the files to be resynchronized. If the number of matching lines in the files is less than this number, the FC command displays the matching lines as differences. The default value is 2.

/b Compares the files in binary mode. FC compares the two files byte by byte and does not attempt to resynchronize the files after finding a mismatch. This is the default mode for comparing files that have extensions of .EXE, .COM, .SYS, .OBJ, .LIB, or .BIN.

Notes

Reporting differences between files for an ASCII comparison

When you use FC for an ASCII comparison of two files, DOS reports the differences by displaying lines that do not match between those that do for each file. For example:

```
Comparing files filename1 and filename2
***** filename1
LINE MATCHES
LINE DOES NOT MATCH
LINE DOES NOT MATCH
LINE MATCHES
***** filename2
LINE MATCHES
LINE DOES NOT MATCH
LINE MATCHES
```

Using the /b switch for binary comparisons

DOS uses the following format to report mismatches found during a binary comparison:

```
xxxxxxxx: yy zz
```

The value of *xxxxxxxx* specifies the relative hexadecimal address for the pair of bytes, measured from the beginning of the file. Addresses start at 00000000; the hexadecimal values for *yy* and *zz* represent the mismatched bytes from *filename1* and *filename2*, respectively.

Using wildcards

You can use wildcards (? and *) in either of the file names you specify with the FC command. If you use a wildcard in *filename1*, FC compares all the specified files to the file specified by *filename2*. If you use a wildcard in *filename2*, FC uses the corresponding value from *filename1*.

How FC uses memory

When comparing ASCII files, FC uses an internal buffer (large enough to hold 100 lines) as storage. If the files are larger than the buffer, FC compares what it can load into the buffer. If FC does not find a match in the loaded portions of the files, it stops and displays the following message:

```
Resynch failed. Files are too different.
```

When comparing binary files that are larger than available memory, FC compares both files completely, overlaying the portions in memory with the next portions from the disk. The output is the same as that for files that fit completely in memory.

Examples

Suppose you want to make an ASCII comparison of two text files that are named MONTHLY.RPT and SALES.RPT, and you want to display the results in abbreviated format. To make this comparison, type the following command:

```
fc /a monthly.rpt sales.rpt
```

To make a binary comparison of two batch files named PROFITS.BAT and EARNINGS.BAT, type the following command:

```
fc /b profits.bat earnings.bat
```

The results of this command will be similar to the following:

```
00000002: 72 43
00000004: 65 3A
0000000E: 56 92
00000012: 6D 5C
00000013: 0D 7C
00000014: 0D 0A
00000015: 0A 0D
0000001E: 43 7A
0000001F: 09 0A
00000022: 72 44
     ...
     ...
     ...
000005E0: 00 61
000005E1: 00 73
000005E2: 00 73
000005E3: 00 69
000005E4: 00 67
000005E5: 00 6E
000005E6: 00 6D
000005E7: 00 65
000005E8: 00 6E
FC: EARNINGS.BAT longer than PROFITS.BAT
```

If the PROFITS.BAT and EARNINGS.BAT files were identical, FC would display the following message:

```
FC: no differences encountered
```

To compare every .BAT file in the current directory to the file NEW.BAT, type the following command:

```
fc *.bat new.bat
```

To compare the file NEW.BAT on drive C to the file NEW.BAT on drive D, type the following command.

```
fc c:new.bat d:*.bat
```

To compare each batch file in the root directory on drive C to the file with the same name in the root directory on drive D, type the following command:

```
fc c:*.bat d:*.bat
```

FCBS

Specifies the number of file control blocks (FCBs) that DOS can have open at the same time.

A file control block is a data structure that stores information about a file.

Type

CONFIG.SYS

Syntax

fcbs=*x*

Parameters

x Specifies the number of file control blocks that DOS can have open at one time. Valid values for *x* are in the range 1 through 255. The default value is 4.

Notes

Limitation on opening files

If a program tries to open more than *x* files by using file control blocks, DOS might close the files that were opened earlier.

Recommended use of the FCBS command

You should use the FCBS command only if a program requires you to do so. Most newer programs do not require file control blocks. However, some older programs might require you to use the FCBS command in your CONFIG.SYS file.

Examples

To specify that DOS can have up to eight file control blocks open at the same time, add the following line to your CONFIG.SYS file:

```
fcbs=8
```

FDISK

Starts the Fdisk program, which configures a hard disk for use with DOS.

Fdisk displays a series of menus to help you partition your hard disk for DOS.

Type

DOS, External

Syntax

fdisk [**/status**]

Switches

/status Displays an overview of your hard disk partition information without starting the Fdisk program.

Notes

Using Fdisk to partition a hard disk

You can use Fdisk for the following tasks:

- Creating a primary DOS partition
- Creating an extended DOS partition
- Changing the active partition
- Deleting a partition
- Displaying partition data
- Selecting the next hard disk for partitioning, if a system has multiple hard disks

Changing the size of a partition

To change the size of a partition, you must actually delete the partition and create a new one with a different size.

Maximum partition size

The maximum partition size is 2 gigabytes. Depending on the BIOS of your computer, your system might *not* be able to reach this limit.

CAUTION:
Deleting a partition deletes all the data stored on that partition.

Using FDISK with an assigned drive

FDISK does not work on a drive formed by using the ASSIGN, SUBST, or JOIN command.

FILES

Sets the number of files that DOS can access at one time.

Type

CONFIG.SYS

Syntax

files=*x*

Parameters

x Specifies the number of files that DOS can access at one time. Valid values for *x* are in the range 8 through 255. The default value is 8.

Notes

Although the default setting for the *x* parameter is 8, some programs require a larger value. A typical setting is 20.

Examples

To specify that DOS can access up to 20 files at one time, add the following line to your CONFIG.SYS file:

```
files=20
```

FIND

Searches for a specific string of text in a file or files.

After searching the specified files, FIND displays any lines of text that contain the specified string.

Type

DOS, External, Network

Syntax

find [**/v**][**/c**][**/n**][**/i**]"*string*" [[*drive*:][*path*]*filename*[...]]

Parameters

"*string*"

Specifies the group of characters you want to search for. You must enclose the text for *string* in quotation marks.

[*drive*:][*path*]*filename*

Specifies the location and name of the file in which to search for the specified string.

Switches

/v Displays all lines not containing the specified string.

/c Displays only a count of the lines that contain the specified string.

/n Precedes each line with the line number of the file.

/i Specifies that the search is not to be case-sensitive.

Notes

Specifying a string

Unless you specify the **/i** switch, the Find program searches for exactly what you specify for *string*. For example, to the FIND command, the characters "a" and "A" are different. If you were to use the **/i** switch, however, the case is ignored and the search for "a" and "A" is made as if they were the same character.

If the string you want to search for contains quotation marks, you must use two quotation marks for each quotation mark contained within the string.

Interrupting FIND with a Ctrl+C

When the Find program is interrupted using **Ctrl+C**, an error level code of 0 is returned.

Using FIND as a filter

If you omit a file name, the FIND command acts as a filter, taking input from the DOS standard source (usually the keyboard, a pipe, or a redirected file) and displaying any lines that contain the *string* you specified.

Using wildcards with FIND

You cannot use wildcards (* and ?) in file names or extensions that you specify with the FIND command. To search for a string in a set of files you specify with wildcards, you can use the FIND command in a FOR command. (See "FOR" on page 113 for more information.)

Using the /v or /n switch with the /c switch

If you specify the **/c** and **/v** switches in the same command, the Find program displays a count of the lines that do not contain the specified string. If you specify the **/c** and **/n** switches in the same command, the **/n** switch is ignored.

Using FIND in files with carriage returns

The FIND command does not recognize carriage returns. When you use the FIND command to search for text in a file that includes carriage returns, you must limit the search string to text that can be found between carriage returns—that is, a string that is not likely to be interrupted by a carriage return. For example, no match is reported for the string "tax file" wherever a carriage return occurs between the word "tax" and the word "file".

Examples

To display all lines from the file PENCIL.AD that contain the string "Pencil Sharpener", type the following command:

```
find "Pencil Sharpener" pencil.ad
```

To find a string that contains text within quotation marks, you must enclose the entire string in quotation marks and, in addition, use two quotation marks for each quotation mark contained within the string, as shown in the following example:

```
find "The scientists labeled their paper ""for
discussion only.""It is not a final report."
report.doc
```

If you want to search for a set of files, you can use the FIND command with the FOR command. The following command uses this method to search the current directory for files that have the extension .BAT; in each file found, the command searches for the string "PROMPT":

```
for %f in (*.bat) do find "PROMPT" %f
```

Suppose you want to search your hard disk to find and display the file names on drive C that contain the string "CPU". To do this you can use the pipe (|) to direct the results of a DIR command to the Find program, as shown in the following example:

```
dir c:\ /s /b | find "CPU"
```

Before using a pipe for redirection, you should set the *TEMP* environment variable in your AUTOEXEC.BAT file.

Since searches using the FIND command are case-sensitive and since the DIR command produces uppercase output, you must either type the string "CPU" in uppercase letters or use the **/i** switch with the FIND command.

FOR

Runs a specified command for each file in a set of files.

You can use the FOR command within a batch program or directly from the command prompt.

Type

DOS, Batch, Internal

Syntax

To use FOR in a batch program, use the following syntax:

for %%*variable* **in** (*set*) **do** *command* [*command-parameters*]

To use FOR from the command prompt, use the following syntax:

for %*variable* **in** (*set*) **do** *command* [*command-parameters*]

Parameters

%%*variable* or %*variable*
: Represents a replaceable variable. The FOR command replaces %%*variable* (or %*variable*) with each text string in the specified set until the command (specified in the *command* parameter) processes all the files. Use %%*variable* to carry out the FOR command within a batch program. Use %*variable* to carry out FOR from the command prompt.

(*set*)
: Specifies one or more files or text strings that you want to process with the specified command. The parentheses are required. Wildcards may be used.

command
: Specifies the command that you want to carry out on each file included in the specified set.

command-parameters
: Specifies any parameters or switches that you want to use with the specified command (if the specified command uses any parameters or switches).

Notes

Using the in and do keywords

In and **do** are not parameters, but they are required in the FOR command. If you omit either of these keywords, DOS displays an error message.

Using the replaceable variable

To avoid confusion with the batch parameters **%0** through **%9**, you can use any character for *variable* except the numbers 0 through 9. For simple batch programs, a single character such as **%%f** might be all that is necessary.

You can use multiple values for *variable* in complex batch programs to distinguish different replaceable variables. However, you cannot nest (add) multiple FOR commands on the same command line.

Specifying a group of files

The *set* parameter can represent a single group of files or several groups of files. You can use wildcards (***** and **?**) to specify a file set. The following are valid file sets:

```
(*.doc)

(*.doc *.txt *.me)

(jan*.doc jan*.rpt feb*.doc feb*.rpt)

(ar??1991.* ap??1991.*)
```

When you use the command, the first value in *set* replaces *%%variable* (or*%variable*) and DOS carries out the specified command in order to process this value; this continues until DOS has processed all the files (or groups of files) that correspond to the value (or values) in *set.*

Examples

Suppose you want to use the TYPE command to display the contents of all the files in the current directory that have the extension .DOC or .TXT. To do this and to use the replaceable variable *%f,* type the following command at the command prompt:

```
for %f in (*.doc *.txt) do type %f
```

In this example, each file that has the .DOC or .TXT extension in the current directory is substituted for the *%f* variable until the contents of every file are displayed. To use this command in a batch file, you simply replace every occurrence of *%f* with *%%f.* Otherwise, DOS ignores the variable and displays an error message.

DOS supports command switches, pipes, and redirection that you may want to use with the specified command. For example, to redirect the output of the previous example to PRN (the default printer port), you would type the following command:

```
for %f in (*.doc *.txt) do type %f > prn:
```

Related Information

For additional information about working with batch programs, see *PC DOS User's Guide.*

FORMAT

Formats the disk in the specified drive to accept DOS files.

The FORMAT command creates a new root directory and file allocation table for the disk. It can also check for bad areas on the disk, and it can delete all data on the disk. In order for DOS to be able to use a new disk, you must first use this command to format the disk.

Type

DOS, External

Syntax

format *drive*: [**/v**[:*label*]] [**/q**] [**/u**] [**/f**:*size*] [**/b**|**/s**]

format *drive*: [**/v**[:*label*]] [**/q**] [**/u**] [**/t**:*tracks***/n**:*sectors*] [**/b**|**/s**]

format *drive*: [**/v**[:*label*]] [**/q**] [**/u**][**/1**][**/4**] [**/b**|**/s**]

format *drive*: [**/q**] [**/u**][**/1**][**/4**] [**/8**][**/b**|**/s**]

Parameters

drive: Specifies the drive containing the disk you want to format. If you do not specify any of the following switches, FORMAT uses the drive type to determine the default format for the disk.

Switches

/v:*label* Specifies the volume label. A volume label identifies the disk and can be a maximum of 11 characters. If you omit the **/v** switch, or use it without specifying a volume label, DOS prompts you for the volume label after the formatting is completed. If you format more than one disk by using one FORMAT command, all of the disks will be given the same volume label. The **/v** switch is not compatible with the */8* switch. For more information about disk volume labels, see commands "DIR" on page 72, "LABEL" on page 143, and "VOL" on page 236.

/q Deletes the file allocation table (FAT) and the root directory of a previously formatted disk, but does not scan the disk for bad areas. You should use the **/q** switch to format only previously formatted disks that you know are in good condition.

/u Specifies an unconditional format operation for a diskette or hard disk. Unconditional formatting destroys all existing data on a disk and prevents you from later "unformatting" the disk. You should use **/u** if you have received read and write errors during use of the disk. For information about unformatting a disk, see the command "UNFORMAT" on page 233.

/**f**:*size* Specifies the size of the diskette to format. When possible, use this switch instead of the /**t** and /**n** switches. Use one of the following values for *size*:

160 or **160K** or **160KB**
160K, single-sided, double-density, 5.25-inch diskette

180 or **180K** or **180KB**
180K, single-sided, double-density, 5.25-inch diskette

320 or **320K** or **320KB**
320K, double-sided, double-density, 5.25-inch diskette

360 or **360K** or **360KB**
360K, double-sided, double-density, 5.25-inch diskette

720 or **720K** or **720KB**
720K, double-sided, double-density, 3.5-inch diskette

1200 or **1200K** or **1200KB** or **1.2** or **1.2M** or **1.2MB**
1.2-MB, double-sided, quadruple-density, 5.25-inch diskette

1440 or **1440K** or **1440KB** or **1.44** or **1.44M** or **1.44MB**
1.44-MB, double-sided, quadruple-density, 3.5-inch diskette

2880 or **2880K** or **2880KB** or **2.88** or **2.88M** or **2.88MB**
2.88-MB, double-sided, 3.5-inch diskette

/**b** Reserves space for the system files IBMBIO.COM and IBMDOS.COM on a newly formatted disk (as hidden files). In previous versions of DOS, it was necessary to reserve this space before using the SYS command to copy the system files to the disk. This switch is maintained in DOS for compatibility reasons only.

/**s** Copies the operating system files IBMBIO.COM, IBMDOS.COM, and COMMAND.COM from the startup drive of your system to the newly formatted disk. If the Format program cannot find the operating system files, it prompts you to insert a system diskette.

/**t**:*tracks*
Specifies the number of tracks on the disk. When possible, use the /**f** switch instead of this switch. If you use the /**t** switch, you must also use the /**n** switch. These two switches provide an alternative method of specifying the size of the disk being formatted. You cannot use the /**f** switch with the /**t** switch.

/**n**:*sectors*
Specifies the number of sectors per track. When possible, use the /**f** switch instead of this switch. If you use the /**n** switch, you must also use the /**t** switch. These two switches provide an alternative method of specifying the size of the disk being formatted. You cannot use the /**f** switch with the /**n** switch.

/**1** Formats a single side of a diskette.

/**4** Formats a 5.25-inch, 360K, double-sided, double-density diskette on a 1.2 MB disk drive. Some 360K drives cannot reliably read diskettes formatted with this switch. When used with the /**1** switch, this switch formats a 5.25-inch, 180K, single-sided diskette.

/8 Formats a 5.25-inch diskette with 8 sectors per track. This switch formats a diskette to be compatible with DOS versions prior to 2.0.

Notes

Typing a volume label

After formatting a diskette, the following message is displayed:

```
Volume label (11 characters, ENTER for none)?
```

The volume label can be a maximum of 11 characters (including spaces). If you do not want your diskette to have a volume label, just press **Enter**. For information about volume labels, see command "LABEL" on page 143.

Formatting a hard disk

When you use the FORMAT command to format a hard disk, DOS displays the following *before* attempting to format the hard disk:

```
WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE x: WILL BE LOST!
Proceed with Format (Y/N)?_
```

To format the hard disk, press **Y**; if you do not want to format the disk, press **N**.

Format messages

When formatting is complete, DOS displays messages showing the total disk space, any space marked as defective, the total space used by the operating system (if you used the /**s** or /**b** switch), and the space available for your files.

Safe formatting

If you do not specify the /**u** switch or a switch that reformats the disk to a different size, the Format program performs a "safe" format, meaning that it clears the file allocation table and root directory of the disk but does not delete any data. You can then use the UNFORMAT command to recover the disk if you did not originally intend to format the disk. The Format program also checks each sector on the disk to ensure that the sector can properly store data. If it locates a sector that cannot store data, that sector is marked to prevent DOS from using it.

If you specify the /**u** switch or any switch that changes the size of the disk, an unconditional format operation is processed by deleting all data on the disk.

Quick formatting

You can speed up the formatting process by using the /**q** switch. Use this switch only if you have not received read or write errors on your hard disk. You can speed up the process even more by using both the /**q** and /**u** switches. If you use the /**u** switch, the information necessary to later unformat the disk is not saved..

Formatting a new disk

When you use the Format program to format a disk that has never been formatted, specify the **/u** switch to minimize formatting time.

Using FORMAT with a reassigned drive or a network drive

You should not use the FORMAT command on a drive prepared by using the ASSIGN, JOIN, or SUBST command. You cannot format disks over a network.

FORMAT exit codes

The following list shows each exit code and a brief description of its meaning:

0 The format operation was successful.

3 The user pressed **Ctrl+C** to stop the process.

4 A fatal error occurred (any error other than 0, 3, or 5).

5 The user pressed **N** in response to the prompt "Proceed with Format (Y/N)?" to stop the process.

You can check these exit codes by using the *errorlevel* condition with the IF batch command. For an example of a batch program that supports *errorlevel* conditions, see command "IF" on page 126.

Examples

To format a new diskette in drive A, using the default size, type the following command:

```
format a:
```

To perform a quick format operation on a previously formatted diskette in drive A, type the following command:

```
format a: /q
```

To format a diskette in drive A, completely deleting all data on the disk, type the following command:

```
format a: /u
```

To format a 360K diskette in drive A and copy the operating system to it, type the following command:

```
format a: /f:360 /s
```

To format a diskette in drive A and assign it the volume label "DATA", type the following command:

```
format a: /v:DATA
```

Related Information

For information about restoring disks after using the FORMAT command, see command "UNFORMAT" on page 233.

GOTO

Directs DOS to a line in a batch program marked by a label you specify.

The GOTO command directs DOS within a batch program to a line identified by a label. When DOS finds the label, it processes the commands beginning on the next line.

Type

Batch, Internal

Syntax

goto *label*

Parameters

label Specifies the line in a batch program to which DOS should go.

Notes

Valid values for *label*

The *label* parameter can include spaces but cannot include other separators, such as semicolons or equal signs.

GOTO uses the first eight characters of each label

The GOTO command uses only the first eight characters of a label. Therefore, the labels "hithere01" and "hithere02" are both equivalent to "hithere0".

Matching the *label* parameter with the label in the batch program

The *label* value you specify on the GOTO command line must match a label in the batch program. The label within the batch program must begin with a colon.

If your batch program does not contain the label that you specify, the batch program stops and DOS displays the following message:

```
Label not found
```

DOS recognizes a batch-program line beginning with a colon (:) as a label and does not process it as a command. If a line begins with a colon, DOS ignores any commands on that line.

Using GOTO for conditional operations

The GOTO command is often used on the same command line with other commands to perform conditional operations. For more information about using GOTO for conditional operations, see command "IF" on page 126.

Examples

The following batch program formats a diskette in drive A as a system diskette. If the operation is successful, the GOTO command directs DOS to a label named "end".

```
echo off
format a: /s
if not errorlevel 1 goto end
echo An error occurred
during formatting.
:end
echo End of batch program.
```

Related Information

For additional information about working with batch programs, see the *PC DOS User's Guide.*

GRAPHICS

Loads a program into memory that allows DOS to print graphics.

The GRAPHICS command supports the CGA, EGA, and VGA graphics display modes.

Type

DOS, External, Network

Syntax

graphics [*type*] [[*drive*:][*path*]*filename*] [**/r**] [**/b**] [**/lcd**] [**/printbox:std**|**/printbox:lcd**]

Parameters

type Specifies the type of printer. The following list shows each valid value for this parameter and a brief description of its meaning:

color1	An IBM Personal Computer Color Printer with black ribbon
color4	An IBM Personal Computer Color Printer with RGB (red, green, blue, and black) ribbon

color8	An IBM Personal Computer Color Printer with CMY (cyan, magenta, yellow, and black) ribbon
hpdefault	Any Hewlett-Packard** PCL printer
deskjet	A Hewlett-Packard DeskJet** printer
graphics	An IBM Personal Computer Graphics Printer, IBM Proprinter*, or IBM Quietwriter* printer
graphicswide	An IBM Personal Computer Graphics Printer with an 11-inch-wide carriage, IBM Proprinters II and III XL
laserjet	A Hewlett-Packard LaserJet** printer
laserjetii	A Hewlett-Packard LaserJet II printer
paintjet	A Hewlett-Packard PaintJet** printer
quietjet	A Hewlett-Packard QuietJet printer
quietjetplus	A Hewlett-Packard QuietJet Plus printer
ruggedwriter	A Hewlett-Packard RuggedWriter printer
ruggedwriterwide	A Hewlett-Packard RuggedWriterwide printer
thermal	An IBM PC-convertible Thermal Printer
thinkjet	A Hewlett-Packard ThinkJet printer

[*drive*:][*path*]*filename*

Specifies the location and name of the printer profile that contains information about all supported printers. If this parameter is omitted, DOS looks for a file called GRAPHICS.PRO in the current directory and in the directory that contains the GRAPHICS.COM file.

Switches

/r
Prints the image as it appears on the screen (white characters on a black background) rather than reversed (black characters on a white background). The latter occurs by default.

/b
Prints the background in color. This switch is valid for *color4* and *color8* printers.

/lcd
Prints an image by using the liquid crystal display (LCD) aspect ratio instead of the CGA aspect ratio. The effect of this switch is the same as that of **/printbox:lcd**.

/printbox:std|/printbox:lcd

Selects the print-box size. You can abbreviate **printbox** as **pb**. You should check the first operand of the **printbox** statement in your GRAPHICS.PRO file and specify the **/printbox:std** switch if that operand is **std** or the **/printbox:lcd** switch if that operand is **lcd**.

Notes

Printing the contents of the screen

To print the contents of the screen, press the **Shift+Print Screen** key combination. If the computer is in 320 x 200 color graphics mode and if the printer type is *color1* or *graphics*, the GRAPHICS command prints the screen contents with as many as four shades of gray. If the computer is in 640 x 200 color graphics mode, the screen contents are printed sideways on the paper (landscape orientation). You cannot use the **Shift+Print Screen** key combination to print the contents of a screen to a PostScript** printer.

Effect on memory

The GRAPHICS command decreases the amount of available conventional memory.

Loading a new profile

If you have already loaded a printer profile and you want to load another one by using the GRAPHICS command, the new profile must be smaller than the one already loaded.

To load a new profile that is larger than the one currently loaded, you must restart your system and then use the GRAPHICS command to load the new profile.

If you try to use only the GRAPHICS command to load a new profile that is larger than the currently loaded profile, DOS displays the following message:

```
Unable to reload with profile supplied
```

Examples

To prepare to print a graphics screen on your printer, type the following command:

```
graphics
```

After you display the information you want to print, press **Shift+Print Screen**. DOS scans the information displayed on the screen and sends it to the printer.

Related Information

For information about printing text files, see command “PRINT” on page 188.

HELP

Provides online information about the DOS commands.

The information that this command displays is similar to, but less detailed than, that found in this chapter.

Type

DOS, External, Network

Syntax

help [*command*]

Parameters

command

Specifies the name of the command about which you want information. If you do not specify a command name, the HELP command lists and briefly describes every command provided with DOS.

Notes

There are two ways to get online help for a command. You can specify the name of the command on the HELP command line, or you can type the name of the command and the **/?** switch at the command prompt. For example, you can type either of the following commands to get information about the XCOPY command:

```
help xcopy

xcopy /?
```

The second command is slightly faster.

HIMEM.SYS

This is one of the installable device drivers provided by DOS. It manages the use of extended memory on a computer with at least an 80286 processor and extended memory. DOS Setup installs this device driver automatically on such systems.

Related Information

For a detailed description of this device driver, see “HIMEM.SYS” on page 263.

IBMAVD

This command opens the IBM AntiVirus/DOS full-screen utility. Follow the instructions to initiate a check of your hard disk and diskette files for known viruses. If a virus is detected, the utility notifies you and helps you remove the virus.

Type

DOS, External

Syntax

ibmavd

IBMAVW

Starts the Windows anti-virus program.

Type

DOS, Windows

Syntax

From the DOS command-line prompt:

win ibmavw

From the Windows desktop: Double-click on the IBM AntiVirus Icon in the IBM Tools folder.

IBMAVSP

Starts IBMAVSP, the IBM AntiVirus stand-alone program. This program can help you find and clean up a virus when your system is too badly infected to use the full-screen IBM AntiVirus program. It can also be incorporated into BAT files.

If no options are specified, IBMAVSP operates interactively.

Type

DOS, External

Syntax

ibmavsp [*][*n][*drive*...][-**mem**] [-**allfiles**][-**programs**][-**oneflop**]
[-**log***logfilename*][-**vlog**][-**nlog**][-**nb**][-**copenerr**][-**cerr**] [-**nrep**][-**nwipe**][-**nfscan**][-**yrep**]
[-**ywipe**][-**yfscan**]

Parameters

***** Scans all local fixed drives for viruses. If this option is used, the IBMAVSP program will not ask you what drives to scan.

***n** Scans all network drives. If this option is used, IBMAVSP will not ask you what drives to scan.

drive:... Scans the drive specified. One or more drive letters can be specified on the command line. When drive letters are specified, IBMAVSP will not ask you what drives to scan.

Switches

-mem Scans memory. This may be used when only memory (not drives) needs to be checked. If this option is used, IBMAVSP will not ask you what drives to scan.

-allfiles Scans all files on the specified drives for viruses. If this option is used, IBMAVSP will not ask you whether all files, or just programs, should be scanned.

-programs Scans only programs on the specified drives for viruses. If this option is used, IBMAVSP will not ask you whether all files, or just programs, should be scanned.

-oneflop Specifies that the IBMAVSP program does not ask you if a second diskette should be scanned.

-log*logfilename* Specifes where IBMAVSP is to place the log file. If *logfilename* is not specified, the log is written to the file IBMAVSP.LOG in the current directory. If *logfilename* is a simple file name, the log will be written to the current directory. If it is a fully-qualified file name, the log will be written to the specified drive and directory. If this option is used, IBMAVSP will not ask you where the log file should be placed.

-vlog Specifies that the names of all scanned files and boot sectors are placed into the log file, regardless of whether an object was infected or not.

-nlog Specifies that a log file will not be created.

-nb Specifies no beep when a virus is found.

-copenerr Specifies to continue checking for viruses without asking you, even if a file cannot be opened for scanning.

-cerr Specifies to continue checking for viruses without asking you, if a nonfatal error is encountered.

-nrep Specifies that infected files or boot sectors should not be repaired. If this option is used, IBMAVSP will not ask you if infected files or boot sectors should be repaired.

-nwipe Specifies that infected files should not be erased and infected boot sectors should not be replaced if they cannot be repaired. If this option is used, IBMAVSP will not ask you if these objects should be erased or replaced.

-nfscan Specifies not to offer to perform a final thorough scan, if viruses are found.

-yrep Specifies that infected files or boot sectors are to be repaired. If this option is used, IBMAVSP will not ask you if infected files or boot sectors should be repaired.

-ywipe Specifies that infected files should be erased and infected boot sectors should be replaced if they cannot be repaired. If this option is used, IBMAVSP will not ask you if these objects should be erased or replaced.

-yfscan Specifies that a final, thorough scan should be performed automatically if viruses are found.

Examples

To operate the IBMAVSP interactively, type the following command and answer the prompts.

```
ibmavsp
```

To tell IBMAVSP to scan all programs on all local hard disk drives and place its log file in the file "c:\ibmavsp.log", type the following command. This method of operation is useful in .BAT files, and whenever your interaction is not desired.

```
ibmavsp * -programs -splc:\ibmavsp.log
```

For more information about how this program works see the *PC DOS User's Guide*.

IF

Performs conditional processing in batch programs.

If the condition specified in an IF command is true, DOS carries out the command that follows the condition. If the condition is false, DOS ignores the command.

Type

Batch, Internal

Syntax

if [*not*] **errorlevel** *number command*

if [*not*] *string1==string2 command*

if [*not*] **exist** *filename command*

Parameters

not Specifies that DOS should carry out the command only if the condition is false.

errorlevel *number*
Specifies a true condition only if the previous program run by COMMAND.COM returned an exit code equal to or greater than *number.*

command
Specifies the command that DOS should carry out if the preceding condition is met.

string1==string2
Specifies a true condition only if *string1* and *string2* are the same. These values can be literal strings or batch variables (*%1,* for example). Literal strings do not need quotation marks.

exist *filename*
Specifies a true condition if *filename* exists.

Notes

When a program stops, it returns an exit code to DOS. The **errorlevel** parameter lets you use exit codes as conditions.

Examples

The following example displays the message "Cannot find data file" if DOS cannot find the file PRODUCT.DAT:

```
if not exist product.dat echo Cannot find data file
```

The following example displays an error message if an error occurs during formatting of the diskette in drive A. If no error occurs, the error message is skipped.

```
:begin
echo off format a: /s
if not errorlevel 1 goto end
echo An error occurred during formatting.
:end
echo End of batch program.
```

The following example tests for the existence of a directory. The IF command cannot be used to test directly for a directory, but the null (NUL) device does exist in every directory. Therefore, you can test for the null device to determine whether a directory exists.

```
if exist c:\mydir\nul goto process
```

Related Information

For additional information about working with batch programs, see the *PC DOS User's Guide.*

INCLUDE

You can include the contents of one configuration block in another by using the INCLUDE command. The INCLUDE command instructs DOS to carry out the commands in another configuration block as well as the commands in the current block. This command specifies the name of the block you want to include; the command can be used only within a configuration block within your CONFIG.SYS file.

Type

CONFIG.SYS

Syntax

include=*blockname*

Parameters

blockname Specifies the name of the configuration block to include.

Examples

Suppose you wanted to add a configuration to a previous set of configuration blocks. You could use the INCLUDE command to do this similar to the following:

```
:
[LOADALL]
include=CPSW
include=DLS
include=INTLNK
set path=c:\net;c:\dos
:

[Common]
```

The [LOADALL] configuration uses the INCLUDE command to include the [CPSW], [DLS], and [Network] blocks. It also has its own SET PATH command.

INSTALL

Loads a memory-resident program into memory when you start DOS.

Memory-resident programs stay in memory as long as your system is on. They can be used even when other programs are active. You can use the INSTALL command to load DOS memory-resident programs such as FASTOPEN, KEYB, NLSFUNC, and SHARE.

Type

CONFIG.SYS

Syntax

install=[*drive*:][*path*]*filename* [*command-parameters*]

Parameters

[*drive*:][*path*]*filename*
Specifies the location and name of the memory-resident program you want to run.

command-parameters
Specifies parameters for the program you specify for *filename.*

Notes

The Install process does not create an environment for a program it loads. Therefore, slightly less memory is used if you load a program using the INSTALL Command rather than from your AUTOEXEC.BAT file. Some programs might not run correctly if they are loaded using the INSTALL command. Do not use the Install process to load programs that use environment variables or shortcut keys or that require COMMAND.COM to be present to handle critical errors.

Examples

Suppose you want to install the FASTOPEN.EXE, located in the DOS directory on drive C, from your CONFIG.SYS file instead of from your AUTOEXEC.BAT file or the command line. In addition, you want to specify that the Fastopen program track the opening of up to 100 files and directories on drive C. To do this, include the following command in your CONFIG.SYS file:

```
install=c:\dos\fastopen.exe c:=100
```

You can use the INSTALLHIGH= command in the CONFIG.SYS file, if you want to load the command in the high memory area of your computer.

INSTALLHIGH

Loads a terminate-and-stay-resident program into upper memory blocks. If upper memory is not available, the INSTALLHIGH command functions just like the INSTALL command.

You can use this command only in your CONFIG.SYS file.

Type

CONFIG.SYS

Syntax

installhigh=[*drive*:] [*path*]*filename* [*command-parameters*]

Parameters

[*drive*:][*path*]*filename*
Specifies the location and name of the memory-resident program you want to run.

command-parameters
Specifies parameters for the program you specify for *filename*.

Notes

No environment created

The Installhigh process does not create an environment for a program it loads. Therefore, slightly less memory is used if you load a program using the INSTALLHIGH command rather than from your AUTOEXEC.BAT file. Some programs might not run correctly if they are loaded using the INSTALLHIGH command. Do not use the Installhigh process to load programs that use environment variables or shortcut keys or that require COMMAND.COM to be present to handle critical errors.

DOS=UMB command required

To use the INSTALLHIGH command, you must include the DOS=UMB command in your CONFIG.SYS file.

Upper-memory manager must be installed

Before you can use the Installhigh program to load a program into the upper memory area, you must install an upper memory area manager. DOS provides the EMM386.EXE device driver to manage the upper memory area for computers with an 80386 or higher processor. For other computers, you may be able to use the following device drivers:

- UMBMONO.SYS
- UMBCGA.SYS
- UMBHERC.SYS
- UMBEMS.SYS

Affect of RAMBoost on the INSTALLHIGH command

If the RAMBoost program is running or if you start the RAMBoost program, the INSTALLHIGH command is ignored. Programs are analyzed by RAMBoost and positioned where RAMBoost determines is best for your computer. For more information about the RAMBoost program, see the *PC DOS User's Guide* and “RAMSETUP” on page 194.

Related Information

For more information about device drivers, see Chapter 5, "Working With Device Drivers" on page 245.

INTERLNK

The INTERLNK command starts the file transfer utility program that allows you to connect two computers using the serial ports or parallel ports through a client-server set-up. The server computer runs the file transfer program. Once the connection is made to the server computer, the client computer uses devices on the server computer as though they were local devices.

Type

DOS, Network

Syntax

To run the client program from the command line:

interlnk [*client*:=[*server*:]]

To exclude loading specific driver functions, type the following in your CONFIG.SYS file:

device=*drive*:\*path*\interlnk [**/drives**:*n*] [**/noprinter**] [**/auto**]

Parameters

client: Specifies a local drive to map to a remote drive. The local drive must be created by the InterLnk program. It cannot be a physical drive on the local computer. The mapping is disconnected if no remote drive is specified.

server: Specifies a remote drive to map to a local drive.

drive:\path
Indicates the location of the InterLnk program.

Switches

/drives:*n* Specifies the maximum number of drives to redirect. The default is 3.

/noprinter Specifies that printer redirection is not to be enabled.

/auto Specifies that the device driver is not to be loaded unless a connection is made at startup.

Notes

Using INTERLNK.EXE

Before you can use the file transfer capability of the Interlnk program, you must have the program installed on both the client and server computers.

General overview

The InterLnk program does not generally require any special switches or parameters. Serial and parallel communications support and printer support are installed by default. Hardware ports and interrupt levels are set automatically. The InterLnk program tries to load itself into the upper memory blocks when space is available.

Examples

To show the status of the InterLnk program, type:

```
interlnk
```

If the InterLnk program is not installed on your system, a message is displayed informing you that you need to add the following line to your CONFIG.SYS file:

```
    device=c:\dos\interlnk.exe
```

For additional information see the *PC DOS User's Guide*.

Related Information

For more information about the file transfer server, see command “INTERSVR.”

For information about working with device drivers, see Chapter 5, “Working With Device Drivers” on page 245.

INTERSVR

Provides serial or parallel file transfer and printing capabilities by means of redirected drives.

Type

DOS, Network

Syntax

To set up the server, use:

intersvr [*drive*:[...]] [/**x**=*drive*:[...]] [/**lpt**[:][*n*|*address*]] [/**com**[:] [*n*|*address*]] [/**baud**:*rate*][/**v**][/**b**]

To copy Interlnk files from one computer to another, provided that the computers are connected by means of their serial ports with a 7-wire null-modem cable, use:

intersvr /rcopy

Parameters

drive: Specifies the drive(s) to redirect.

Switches

/x=*drive*:
Specifies the drive(s) to be excluded from redirection.

/lpt:*n* Specifies which port to scan (**/lpt** scans all LPT ports).

/lpt:*address*
Specifies which port address to scan.

/com:*n* Specifies which port to scan (**/com** scans all serial ports).

/com:*address*
Specifies which port address to scan.

/baud:*rate*
Sets a maximum serial baud rate. Valid values are: 9600, 19200, 38400, 57600, and 115200 baud.

/v Forces the Intersvr program to drop into the variable mode if using a serial connection.

/b Displays the Interlnk server screen in black and white.

/rcopy Copies Interlnk files from one computer to another. When the **/rcopy** switch is used, no other switch can be specified except **/b**, and it must come before the **/rcopy** switch.

Notes

Required installation

Before you can use the file transfer capability of the Interlnk program, you must have the program installed on both the client and the server computers.

You can use the /**rcopy** switch to do this, if the computers are connected with serial cables.

How the operation of the Advanced Power Manager (APM) is affected

The INTERSVR command disables APM, if it is installed on your system. APM (or POWER.EXE) is used to conserve power by placing inactive operations in the suspend mode. For more information about APM, see "POWER" on page 186. The Intersvr program cannot access a computer after it is in the suspend mode.

Using INTERLNK.EXE and INTERSVR.EXE

Before you can use the file transfer capability of the Interlnk program, you must have the program installed on both the client and the server computers. You can use the **/rcopy** switch to do this, if the computers are connected with serial cables. You can do this as follows:

1. Type the following at the server computer command prompt:

   ```
   intersvr /rcopy
   ```

   The Interlnk Remote Installation screen is displayed. Follow the directions on the screen to complete the data transfer.
2. Restart the client computer (press **Ctrl+Alt+Del**).

Note: The remote install option cannot be used with parallel connections.

Starting INTERLNK and INTERSVR

Use the following steps to start the file transfer utility programs.

1. Decide which computer you want to be the server computer. At the command prompt on the server computer, type the following:

   ```
   c:\dos\intersvr
   ```

   To use the parallel connection, type the following:

   ```
   c:\dos\intersvr /lpt:1
   ```

2. Add the following statement to the CONFIG.SYS file on the client computer to set up the client computer.

   ```
   device=c:\dos\interlnk.exe
   ```

3. Restart the client computer (press **Ctrl+Alt+Del**).
4. Type the following at the command prompt of the client computer (which is the computer not being used as the server computer) to view the status of the connections:

   ```
   c:\dos\interlnk
   ```

The following screen is displayed:

```
PORT=COM1

This Computer              Other Computer
 (Client)                      (Server)
_______________          ____________________

E: equals                    A:
F: equals                    B:
G: equals                    C:  (33MB)  DOS VOL
LPT3:  equals               LPT1:
```

Transferring data

You can use any DOS commands to look at or copy files from the server computer to the client computer or from client to server. The redirected drives on the server computer are displayed as local drives on the client computer.

To copy a file named MEMO.DOC from Drive G (Drive C on the server computer) to Drive D, type the following at the command prompt of the client computer.

```
copy g:\memo.doc d:
```

Mapping Remote Drives to Local Drives

You can change the mapping of any remote drive to any logical drive that exists on the client computer. For example, suppose the client computer has the following drive usage:

```
PORT=COM1

This Computer           Other Computer
 (Client)                  (Server)
_____________          _______________

E: equals                 A:
F: equals                 B:
G: equals                 C:  (33MB)  DOS VOL
LPT3:  equals            LPT1:
```

To access drive A of the server computer as drive G on the client computer, type the following at the command prompt of the client computer:

```
interlnk g=a
```

The following screen is displayed.

```
PORT=COM1

This Computer           Other Computer
 (Client)                  (Server)
_____________          _______________

F: equals                 B:
G: equals                 A:
LPT3:  equals            LPT1:
```

Note: The previous mapping of drive G of the client computer to drive C of the server computer is removed. If you want to reconnect to drive C of the server computer, remap drive C to a valid local drive of the client computer by using the INTERLNK command.

Excluding Drives from the Server

To exclude a drive on the server computer so the client computer cannot access it, type the following at the server computer command prompt:

```
intersvr /x=c
```

Then, the following screen is displayed.

```
PORT=COM1

This Computer            Other Computer
 (Client)                   (Server)
_______________         ____________________

E: equals                  A:
F: equals                  B:
LPT3:  equals             LPT1:
```

Note: Drive C of the server computer is not displayed on the mapping once the connection is made to the client computer.

Related Information

For more information about the file transfer server, see command "INTERLNK" on page 131.

For more information about the INTERLNK.EXE and INTERSVR.EXE device drivers, see Chapter 5, "Working With Device Drivers" on page 245.

For more information about the Interlnk program and how it works, see the *PC DOS User's Guide*.

JOIN

Logically connects a drive to a directory on another drive to produce a single directory structure from two separate directories.

Type

DOS, External

Syntax

join [*drive1:*[*drive2:*]*path*]

or

join *drive1:* **/d**

Parameters

drive1:	Specifies a disk drive that will appear as a directory on *drive2*.
drive2:	Specifies the drive to be connected to *drive1*.

path

Specifies the directory to which you want to join *drive1*. It must be empty and cannot be the root directory.

Switches

/d

Cancels any previous JOIN commands for the specified drive.

Notes

What is joined by JOIN

The entire tree (starting at the root) of the specified drive is joined, regardless of the current directory of that drive.

JOIN limitations

directories

If the path and directory specified already exists before you use the JOIN command, you cannot use that directory for any other purpose while the JOIN command is in effect. If the directory is not empty, DOS does not complete the join operation and displays the following message:

```
Directory not empty - \directory name
```

If the directory does not exist, DOS tries to create it.

With other commands

The following commands do not work with drives formed by the JOIN command:

ASSIGN
CHKDSK
DISKCOMP
DISKCOPY
FDISK
FORMAT
LABEL
RECOVER
RESTORE
SYS
UNDELETE

Invalid drive

After you use the JOIN command, the drive you specify becomes invalid. For example, if you join drive A, you cannot refer to the drive letter A until you remove the join using the **/d** switch. This means you cannot join the default drive because you will then be on an invalid drive. DOS displays the following message when you try to use a joined drive:

```
Invalid drive specification
```

Network drives

Network drives cannot be specified.

Examples

To see a list of the currently joined drives, use the JOIN command without any parameters. For example, the following message indicates that drive A is joined to the directory path C:\LEVEL1.

```
A:  => C:\LEVEL1
```

You can join any directory or subdirectory in a tree structure. To join the directories FILES and REPORTS on drive C to drive A, you could type:

```
join a:  c:\drivea
```

Directory DRIVEA is created on drive C. If you type `dir c:` for verification, the following is displayed:

```
Volume in drive C is FIXDC
Directory for C:\

.       <DIR>          2:10p
..      <DIR>          2:10p
Files   <DIR>  6-21-93  2:10p
REPORTS <DIR>  7-23-93  5:55p
DRIVEA  <DIR>  8-30-93 12:03a
  5 File(s) 0 bytes
           5287689 bytes free
```

To reverse the previous JOIN command, type the *drive* letter followed by the **/d** switch, as follows:

```
join a: /d
```

Related Information

For information about redirecting disk operations from one drive to another, see command "ASSIGN" on page 24.

For information about substituting a drive letter for a directory name, see command "SUBST" on page 224.

KEYB

Starts the KEYB program, which configures a keyboard for a specific language.

Use KEYB to configure a keyboard for a language other than United States English.

Type

DOS, External, Network

Syntax

keyb [*xx*[,[*yyy*][,[*drive*:][*path*]*filename*]]] [**/e**] [**/id**:*nnn*]

In your CONFIG.SYS file, use the following syntax:

install=[[*dos-drive*:]*dos-path*]**keyb.com** [*xx*[,[*yyy*][,[*drive*:][*path*]*filename*]]] [**/e**] [**/id**:*nnn*]

Parameters

xx Specifies the keyboard code.

yyy Specifies the code page.

[*drive*:][*path*]*filename*
Specifies the location and name of the keyboard definition file. The default filename is KEYBOARD.SYS.

[*dos-drive*:]*dos-path*
Specifies the location of the KEYB.COM file.

Switches

/e Specifies that an enhanced keyboard is installed. Use this switch if you are using an enhanced keyboard with an 8086 computer.

/id:*nnn* Specifies the keyboard in use. This switch is necessary only for countries that have more than one keyboard layout for the same language (for example, France and Italy).

Notes

Values for *xx*, *yyy*, and *nnn*

The following table shows the valid values for *xx*, *yyy*, and *nnn* for each country or language:

Country or language	Keyboard code	Code page	Keyboard identification	Country code
Albanian	al	852, 850	448	355
Australia	us	437, 850		061
Belgium	be	850, 437		032
Bosnia/Herzegovina	yu	852, 850	234	387
Brazil	br	850, 437		055
Bulgaria	bg	855, 850	442, 241	035
Canadian-French	cf	850, 863		002
Croatia	yu	852, 850	234	384
Czech	cz	852, 850	243	042
Denmark	dk	850		045
Finland	su	850, 437		358
France	fr	850, 437	120, 189	033
Germany	gr	850, 437		049
Greece	gk	869, 850		030
Hungary	hu	852, 850	208	036
Iceland	ic	850, 861		354

Country or language	Keyboard code	Code page	Keyboard identification	Country code
International English		437, 850		061
Italy	it	850, 437	141, 142	039
Japan	jp	437, 932		081
Latin America	la	850, 437		003
FYR Macedonia	yc	855, 852	118	389
Netherlands	nl	850, 437		031
Norway	no	850		047
Poland	pl	852, 850	214	048
Portugal	po	850, 860		351
Romania	ro	852, 850	446	040
Russian	ru	866	441, 443, 341 (Russian DOS only)	
Serbia/Montenegro	yc	855, 852	118	381
Slovakia	sl	852, 850	245	042
Slovenia	yu	852, 850	234	386
Spain	sp	850, 437		034
Sweden	sv	850, 437		046
Switzerland (French)	sf	850, 437		041
Switzerland (German)	sg	850, 437		
Thai		,	Keyboard available separately	
Turkey	tr	857, 850	440, 179	090
United Kingdom	uk	850, 437	166, 168	044
United States	us	437, 850		001
Yugoslavia	yu	852, 850	118	038

Installing code pages

The code page you specify for *yyy* must be installed on your system.

Displaying the current keyboard code and code page

If you use the KEYB command with no parameters or switches, DOS lists the current keyboard code, the related code page of the current keyboard, and the current code page used by your console (CON). The information is displayed in the following format:

```
Current keyboard code:  FR code page:  437
Current CON code page:  437
```

Changing the keyboard arrangement

To change the keyboard arrangement, use the KEYB command. You can specify this command from your AUTOEXEC.BAT file or from the DOS command prompt. You can use the KEYB command regardless of whether you have changed any other country-specific settings. The Keyb program works with IBM PC/XT*, IBM PC/AT*, IBM PS/2*, and IBM PC-compatible keyboards.

Using One or more Prepared Code Pages

If the languages you want to use require a prepared code page, you must add at least two commands to your CONFIG.SYS file, and add two or more commands to your AUTOEXEC.BAT file. See the example further in this discussion and the *PC DOS User's Guide* for more information.

Switching between KEYB settings

You can switch from the current KEYB keyboard configuration to the default keyboard configuration at any time by pressing **Ctrl+Alt+F1**. To return to the memory-resident keyboard configuration, press **Ctrl+Alt+F2**. You can switch to "typewriter mode," the standard for some countries, by pressing **Ctrl+Alt+F7**.

Implementing KEYB

The following list shows the three different ways that you can start the Keyb program:

- Type KEYB at the command prompt.
- Include an INSTALL command for KEYB.COM in your CONFIG.SYS file.
- Include the appropriate KEYB command in your AUTOEXEC.BAT file.

KEYB exit codes

The following list shows each exit code and a brief description of its meaning:

0	Keyboard definition file was loaded successfully.
1	Invalid keyboard code, code page, or syntax was used.
2	Keyboard definition file is bad or missing.
4	An error occurred while communicating with the CON device.
5	The requested code page has not been prepared.

You can use the **errorlevel** parameter on the IF command line in a batch program to process exit codes returned by KEYB. For an example of a batch program that processes exit codes, see the DISKCOMP command.

Examples

The following example shows the commands you need to add to your CONFIG.SYS and AUTOEXEC.BAT files to use one or more prepared code pages.

Suppose your current code page defaults to 850 and your hardware code page is 437. However, you want to use code page 850 with a Belgian keyboard and an EGA monitor. If the DOS files you need are in the C:\DOS directory, you can use the following commands in your CONFIG.SYS file to change to Belgian conventions (032) and install the display driver that enables you to switch between code pages:

```
country=032,,c:\dos\country.sys
device=c:\dos\display.sys con=(ega,437,1)
```

The COUNTRY command sets the Belgian conventions for date, time, currency, character sort order, and file name characters. The DEVICE command installs DISPLAY.SYS, indicates that you have an EGA monitor with a 437 hardware code page and reserves space for one prepared code page.

Note: If you were using a VGA monitor, you would put VGA where EGA is shown in the example. Also, if you were using two prepared code pages, you would replace the 1 with a 2.

In your AUTOEXEC.BAT file, you include the following commands to prepare and select code page 850:

```
nlsfunc
mode con cp prep=((850)c:\dos\ega.cpi)
keyb be,,c:\dos\keyboard.sys
chcp 850
```

You include the NLSFUNC command so you can switch between code pages for all devices at the same time. The MODE command loads code page 850 from the EGA.CPI file. The KEYB command changes the arrangement of the keyboard to match a Belgian keyboard. The CHCP command makes code page 850 active.

Notes:

1. The EGA.CPI file is used with code pages 437, 850, 852, 855, 860 and 863.
2. EGAX.CPI is used with code pages 857, 861 and 869.
3. 4201.CPI is used with code page 866.

The following NLSFUNC command can be included in the CONFIG.SYS file rather than in the AUTOEXEC.BAT file:

```
install=c:\dos\nlsfunc.exe
```

You can use the United States keyboard temporarily by pressing **CTRL+ALT+F1**. To return to the Belgian keyboard, you would press **CTRL+ALT+F2**.

Make sure you restart your computer by pressing **CTRL+ALT+DEL** after making changes to your CONFIG.SYS and AUTOEXEC.BAT files.

If you want to use a German keyboard and your KEYBOARD.SYS file is in the DOS directory on drive C, type the following command:

```
keyb gr,,c:\dos\keyboard.sys
```

Related Information

For information about using active and prepared code pages, see command "CHCP" on page 31.

For information about preparing devices for code page switching, see "MODE" on page 158.

For information about the NLSFUNC command and the Nlsfunc program, see "NLSFUNC" on page 173.

For more information about using code pages and the KEYB command, see the *PC DOS User's Guide.*

LABEL

Creates, changes, or deletes the volume label (name) of a disk.

DOS displays the volume label as part of the directory listing. If a volume serial number exists, DOS displays this number as well. For more information about the LABEL command, see the *PC DOS User's Guide.*

Type

DOS, External

Syntax

label [*drive*:][*label*]

To specify that DOS is to display the current volume label and serial number, if they exist, and that DOS is to prompt you to enter a label or delete the existing one, use the following syntax:

label

Parameters

drive: Specifies the location of the disk you want to name.

label Specifies the new volume label. You must include a colon (:) between *drive* and *label.*

Notes

LABEL command messages

If you do not specify a label when you use the LABEL command, DOS displays a message in the following format:

```
Volume in drive A is xxxxxxxxxxx
Volume Serial Number is xxxx-xxxx
Volume label (11 characters, ENTER for none)?
```

The “Volume Serial Number” part of the message is not displayed if the disk has no serial number.

You can type the volume label you want or press **Enter** to delete the current label. If a disk has a label and you press **Enter** for none, DOS prompts you with the following message:

```
Delete current volume label (Y/N)?
```

Press **Y** to delete the label; press **N** to keep the label.

Limitations on volume label names

A volume label can contain as many as 11 characters and can include spaces but no tabs. Consecutive spaces may be interpreted as a single space.

Do not use any of the following characters in a volume label:

```
* ?  / \ | . , ; :  + = [  ]  ( ) & ^  < > "
```

DOS displays volume labels in uppercase letters. If you enter a volume label in lowercase letters, the LABEL command converts the letters to uppercase.

Using LABEL with a redirected drive

The LABEL command does not work on a drive created with the ASSIGN, JOIN, or SUBST command.

Examples

To label a diskette in drive A that contains sales information for 1993, you might type the following:

```
label a:sales1993
```

Related Information

For information about displaying the current disk label, see command “DIR” on page 72 or command “VOL” on page 236.

For information about the volume serial number of a disk, see command “VOL” on page 236.

LASTDRIVE

Specifies the maximum number of drives you can access.

The value you specify represents the last valid drive DOS is to recognize.

Type

CONFIG.SYS

Syntax

lastdrive=*x*

Parameter

x Specifies a drive letter in the range A through Z. The minimum value for this parameter is the letter that corresponds to the number of drives installed on your system. For example, one drive = A, two drives = B, and so on.

Notes

Default setting

The last drive, by default, is the one after the last drive being used on your computer.

Effect on memory

DOS allocates a data structure in memory for each drive specified by the LASTDRIVE command, so you should not specify more drives than are necessary.

Examples

The following command sets the last drive to M, giving your computer access to 13 logical drives:

```
lastdrive=m
```

LOADFIX

Loads a program above the first 64K of memory and runs the program.

Type

DOS, External

Syntax

loadfix [*drive*][*path*]*filename* [program-parameters]

Parameters

drive path
Specifies the drive and directory of the program.

filename
Specifies the name of the program

program-parameters
Specifies any of the programs parameters that you want.

Notes

Packed file corrupt

Use the LOADFIX program to load a program if you receive the message:

```
Packed file corrupt
```

Some programs will cause this message to be displayed when all or a portion of the program has been loaded in the first 64K of conventional memory and cannot run successfully.

Where to use the LOADFIX command

To use the LOADFIX command, include it at the beginning of the command that starts the program. You can type this command from the command prompt or add the command to your AUTOEXEC.BAT file.

LOADHIGH

Loads a program into the upper memory area.

Loading a program into the upper memory area leaves more room in conventional memory for other programs.

Type

DOS, Internal, Network

Syntax

loadhigh [*drive*:][*path*]*filename* [*parameters*]

or

lh [*drive*:][*path*]*filename* [*parameters*]

To specify the regions in memory into which to load the program, use the following syntax:

loadhigh [/l:*region1*[,*minsize1*]][;*region2*[,*minsize2*]...][*drive*:][*path*]*filename* [*parameters*]

Parameters

[*drive*:][*path*]*filename*
Specifies the location and name of the program you want to load.

parameters
Specifies any command-line information required by the program.

Switches

/l:*region1* Specifies the region of memory into which to load the program or a selection of regions that the program can use.. The following parameters can be used with this switch:

,minsize Specifies that DOS load the program into the specified region only if it contains a UMB that is larger than both the load size and the *minsize*.

region2 Specifies an additional region that DOS can use to load a program if the first region cannot be used.

,minsize This parameter has the same effect as when you use it with *region1*

... This indicates that you can keep specifying a region and minsize for as many regions as your computer contains.

Notes

Separating the *minsize* from the region

You must separate the *minsize* parameter from the *region* parameter by using a comma between the values. For example:

```
/L:2,64
```

Separating one region from another

You must separate region values (*region,minsize*) from each other by using a semicolon. For example:

```
/L:2,64;3,64;4
```

Identifying UMB free space

You can use the MEM **/f** command to receive a displayed list of the free areas of memory in your computer. You can also use the MEM **/m** *modulename* command to determine how a particular program uses memory. Using the MEM command in this way helps you to determine the region and minimum size values needed to store a particular program.

DOS=UMB command required

To use the LOADHIGH command, you must include the DOS=UMB command in your CONFIG.SYS file. For more information about the DOS=UMB command, see command "DOS" on page 82.

Upper-memory-area manager must be installed

Before you can load a program into the upper memory area, you must install an upper-memory-area manager. DOS provides EMM386.EXE, which manages the upper memory area for computers with an 80386 or higher processor. (UMBMONO.SYS, UMBCGA, UMBEMS, and UMBHERC can be used as upper memory providers on machines with 8088 or 286 processors.) Before you install EMM386.EXE, you must install the HIMEM.SYS extended-memory manager. Use the DEVICE command in your CONFIG.SYS file to install HIMEM.SYS and EMM386.EXE. These commands must appear before the LOADHIGH command.

How LOADHIGH works

If you use the LOADHIGH command to load a program, DOS attempts to load it into the upper memory area. If there is insufficient space in the upper memory area, DOS loads the program into conventional memory. DOS does not indicate which memory area is used.

Using LOADHIGH in your AUTOEXEC.BAT file

The most convenient way to use the LOADHIGH command is to include it in your AUTOEXEC.BAT file.

The affect of RAMBOOST

If the RAMBoost program is running or if you start the RAMBoost program, the LOADHIGH command is reconfigured. Programs are analyzed by the RAMBoost program and positioned where is is determined to be best for your computer. For more information about the RAMBoost program, see the *PC DOS User's Guide*.

Also, see command "RAMSETUP" on page 194.

Examples

The following command directs DOS to run the MYPROG.EXE program in upper memory, and load the program into upper memory block 2.

```
lh /l:2 c:\myprog.exe
```

Related Information

For information about loading device drivers into the upper memory area, see command "DEVICEHIGH" on page 70.

For information about loading device drivers into conventional memory, see command "DEVICE" on page 69.

MEM

Displays the amount of used and free memory in your system.

You can use the MEM command to display information about allocated memory areas, free memory areas, and programs that are currently loaded into memory.

Type

DOS, External, Network

Syntax

mem [/classify | /free | /debug | /module:*modulename*] [/page]

To display the status of the used and free memory in your system, use the following syntax:

mem

Switches

/classify
Lists the programs that are loaded into memory and shows how much conventional and upper memory each program is using. The MEM **/classify** command also summarizes overall memory use and lists the largest memory blocks available.

You can abbreviate **/classify** as **/c**.

Note: The **/page** switch is the only MEM switch that can be used with **/classify**.

/free Lists the free areas of conventional and upper memory. MEM **/free** shows the segment address and size of each free area of conventional memory and shows the largest free upper memory block in each region of upper memory. MEM **/free** also summarizes overall memory use.

You can abbreviate **/free** as **/f**.

Note: The **/page** switch is the only MEM switch that can be used with **/free**.

/debug List the programs and internal drivers that are currently loaded into memory. MEM **/debug** shows the size, segment address, and type of each module, summarizes overall memory use, and displays other information useful for programming.

You can abbreviate **/debug** as **/d**.

Note: The **/page** switch is the only MEM switch that can be used with **/debug**.

/module:*modulename*
Shows how the specified program module is currently using memory. To display this information, specify the program name after the **/module** switch. MEM **/module** lists the areas of memory the specified program module has allocated, and shows the address and size of each area.

You can abbreviate **/module** as **/m**.

Note: The **/page** switch is the only MEM switch that can be used with **/module**.

/page Pauses after each page of output. This switch can be used with any of the other MEM switches.

You can abbreviate **/page** as **/p**.

Note: The **/page** switch is the only MEM switch that can be used with all the other MEM switches.

Notes

Displaying memory status

DOS displays the status of extended memory only if you have installed memory above the 1-megabyte (MB) boundary in your system. DOS displays the status of expanded memory only if you use expanded memory that conforms to version 4.0 of the Lotus/Intel/Microsoft Expanded Memory Specification (LIM EMS). DOS displays the status of the upper memory area only if use of the upper memory area is enabled.

Allocating extended memory

To allocate Interrupt 15H memory and XMS memory at the same time, use the **/int15** switch when you load the HIMEM.SYS device driver.

Examples

Suppose your system has both expanded memory and extended memory. To display a summary of the total memory of your system—conventional, expanded, extended, and upper—and to display a list of programs currently loaded into memory, type the following command:

```
mem /classify
```

To find out what memory a specific program module has allocated, use the MEM **/module** command. To find out what memory the SMARTDRV module has allocated, type the following:

```
mem /module:smartdrv
```

Some program modules allocate more than one area of memory. The MEM **/module** command displays all the areas of memory allocated by the specified program, shows the segment address and size of each allocation.

Related Information

For information about checking the amount of space available on a disk, see command "CHKDSK" on page 34.

[MENU]

This command must be used when you are going to use menuitems within a multiple configuration block. The [MENU] command defines the items to be displayed on the startup menu.

Type

CONFIG.SYS

Syntax

[menu]

Notes

Importance

To use multiple configuration blocks that involve menu items you must include the [MENU] command as the first line before listing the following related menu item commands:

- INCLUDE
- MENUCOLOR
- MENUDEFAULT
- MENUITEM
- SUBMENU

Using the INCLUDE command

The INCLUDE command is not used exclusively with menu item commands. It can be used to include the contents of one configuration block in another. See "INCLUDE" on page 128 for more information.

For additional information about multiple configuration blocks and how to use menu item commands, see the *PC DOS User's Guide*.

The MENUITEM and SUBMENU commands

A menu block must contain at least one MENUITEM or SUBMENU command.

Examples

To use multiple configurations, you must define a startup menu. To do this, create a configuration block with the block heading [Menu].

When your computer starts, the startup menu appears and lists the available configurations; you choose the configuration you want.

This startup menu block:

```
[MENU]
menuitem=DLS, Load DOS LAN Services Client
menuitem=INTLNK, Load InterLnk Client
menuitem=CPSW, Load Code Page Switching
menucolor=7,1
:
```

Produces this startup menu:

```
PC DOS Startup Menu

1. Load DOS LAN Services Client
2. Load InterLnk Client
3. Load Code Page Switching

Enter a choice:   1
Time remaining:  20
```

In this example:

- The MENUITEM command defines the item on the menu. The first MENUITEM command value, `DLS`, specifies the name of the associated configuration block. The second value, which is optional, specifies the text, `Load DOS LAN Services Client.` to display on the menu. If you do not specify any menu text, DOS uses the name of the configuration block as the menu text.
- The MENUCOLOR command sets the text color to 7 (white) and the background color to 1 (royal blue).
- The MENUDEFAULT command is optional. When this command is used, it specifies which menu item is to be the default configuration. The block must be defined elsewhere in the CONFIG.SYS file. When DOS displays the startup menu, the default menu item is highlighted, and its number appears after the `Enter a choice` prompt. If no item is specified, the default is set to the first item.

Related Information

For information about the MENUCOLOR command, see “MENUCOLOR.”

For information about the MENUDEFAULT command, see “MENUDEFAULT” on page 154.

For information about the MENUITEM command, see “MENUITEM” on page 155

For information about the SUBMENU command, see “SUBMENU” on page 223.

For information about the INCLUDE command, see “INCLUDE” on page 128.

MENUCOLOR

The MENUCOLOR command sets the text and background color for the menu. You can use this command only within a menu block in your CONFIG.SYS file.

Type

CONFIG.SYS

Syntax

menucolor=*x*[,*y*]

Parameters

x Specifies the color of the menu text. You can specify a value from 0 - 15.

,y Specifies the color of the screen background. You can specify a value from 0 - 15.

Notes

Using the *y* parameter

Use of the *y* parameter is optional. If you do not specify a value, DOS defaults the background to black and displays the specified text color on top of it.

If you specify a value, ensure that it is different from the value you specified for the text color otherwise, you may not be able to read the text. If you do specify a value, separate the values with a comma (,).

Color values for 0 - 15

The following colors are represented by the numbers 0 - 15.

Number	Color
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	White
8	Gray
9	Bright Blue
10	Bright Green
11	Bright Cyan
12	Bright Red
13	Bright Magenta

Number	Color
14	Bright Yellow
15	Bright White

Related Information

For information about the [*MENU*] command, see “[MENU]” on page 151.

MENUDEFAULT

This command specifies the default menu item on the startup menu and sets a timeout value if desired. You can use this command only within the menu block in your CONFIG.SYS file.

Type

CONFIG.SYS

Syntax

menudefault= *blockname*[*,timeout*]

Parameters

blockname Specifies the default menu item by its associated configuration block. The block must be defined elsewhere in the CONFIG.SYS file.

,timeout Specifies how many seconds DOS must wait before starting your computer with the default configuration. If you do not specify a timeout value, DOS does not continue until the **Enter** key is pressed.

Notes

Setting a timeout value

The MENUDEFAULT command sets a timeout value. You can specify a timeout value from 0 to 90 seconds. A timeout of 0 seconds forces automatic selection of the default, effectively bypassing the menu display.

If you specify a timeout value but no item is selected within the specified time, DOS starts the computer with the default configuration.

If you do not specify a timeout value, DOS does not continue until the **Enter** key is pressed.

MENUITEM

The MENUITEM command specifies an item on the startup menu. You can use this command only within a menublock in your CONFIG.SYS file.

Type

CONGIG.SYS

Syntax

menuitem= *blockname*[,*menu_text*]

Parameters

blockname Specifies the name of the associated configuration block. The block must be defined elsewhere in the CONFIG.SYS file. If the menu item is selected from the startup menu, DOS carries out the commands in the associated configuration block, as well as any commands at the beginning of the CONFIG.SYS file and any commands in the configuration blocks with the [COMMON] command header.

, *menu_text*
Specifies the text you want DOS to display for the menu item. If you do not specify any menu text, DOS displays the block name as the menu item.

Notes

DOS does not find the blockname

If DOS cannot find the specified blockname, the item does not appear on the startup menu.

Length and markup of the menu text

The menu text can be up to 70 characters long and can contain any characters you want.

Related Information

For more information about menu blocks and their use within multiple configurations, see the *PC DOS User's Guide.*

For more information about the SUBMENU command, see command "SUBMENU" on page 223

For more information about the [MENU] command, see command "[MENU]" on page 151.

MEUTOINI

The MEUTOINI utility program allows you to convert your existing DOS 4.0 Shell data files to the latest version of DOS format. The latest version of DOS Shell uses a single data file (DOSSHELL.INI) rather than multiple DOS 4.0 Shell data files (*.MEU). The DOSSHELL.INI file is created when you install the latest version of DOS. MEUTOINI.EXE converts the old *.MEU files (starting with SHELL.MEU) and adds that information to the new DOS DOSSHELL.INI file.

Type

DOS, External

Syntax

meutoini *drive*:\ *path*

Parameters

drive:*path*

Specifies the location of where your DOS files are located.

Notes

Running MEUTOINI

Type the following to run the MEUTOINI program.:

```
MEUTOINI C:\DOS
```

C:\DOS is the drive and directory from which DOSSHELL is started. This command will copy all of your .MEU files to the DOSSHELL.INI file. If you want to convert only one file named DATA.MEU, type the following:

```
MEUTOINI C:\DOS\DATA.MEU
```

The MEUTOINI.EXE, *.MEU files, and the DOSSHELL.INI file must all be in the directory from which DOSSHELL is started.

Passwords for DOS 4.0 Shell

Passwords for DOS 4.0 Shell files are not copied during this process, and must be re-installed.

MEUTOINI Error Messages

If the MEUTOINI utility program has difficulty converting existing DOS 4.0 Shell data files, it will display one of the following error messages.

Error Message	Description
ERROR: 01	Incorrect number of arguments

Error Message	Description
ERROR: 02	Invalid path or file name
ERROR: 03	Cannot open expected file
ERROR: 04	Cannot open a .TMP file
ERROR: 05	Invalid .MEU file
ERROR: 06*	Nested menu problem
ERROR: 07*	Menu index out of range
ERROR: 08	Cannot open DOSSHELL.INI file
ERROR: 09*	Too many command prompts (greater than 9)
ERROR: 10*	Help is greater than 255 characters
ERROR: 11*	Title is greater than 27 characters
ERROR: 12 <*X*>	Unknown switch in command
ERROR: 13*	Command line is greater than 255 characters

Action: Verify that the command you entered has the correct path and file name, and that all variables in the command are accurate. If the command was entered correctly refer to your Service Information for additional information.

Additional Error Message help

ERRORS 6 and 7 are internal MEUTOINI errors. Record the error number and report the problem. (See the *Service Information Card* for additional information.)

ERRORS 09, 10, 11, and 13 are for your information. Your DOS 4.0 message fields have been truncated and must be re-written.

In ERROR 12 <*X*>, *X* is the unknown switch. *X* can be any letter.

MKDIR (MD)

Creates a directory.

You can use the MKDIR command to create a multilevel directory structure.

Type

DOS, Internal, Network

Syntax

mkdir [*drive*:]*path*

md [*drive*:]*path*

Parameters

drive: Specifies the drive on which you want to create the new directory.

path Specifies the name and location of the new directory. The maximum length of any single path from the root directory to the final directory is 63 characters, including backslashes (\).

Examples

Suppose you want to create a directory on the disk in the current drive and use the directory to store all your tax information. To create a directory named TAXES, type the following command:

```
    mkdir \taxes
```

Now suppose that the TAXES directory is the current directory and that you want to create a subdirectory of TAXES named PROPERTY. To create the PROPERTY directory, type the following command:

```
    mkdir property
```

Related Information

For information about deleting a directory, see command "RMDIR (RD)" on page 203.

For information about changing directories, see command "CHDIR (CD)" on page 33.

MODE

Configures system devices.

The MODE command performs many different tasks, such as displaying system status, changing system settings, or reconfiguring ports or devices.

Type

DOS, External, Network

Syntax

Because the MODE command can perform many different tasks, the syntax necessary to carry out each task is different. Therefore, this reference discusses the tasks separately. For information about the syntax that the Mode program uses for a specific task, see the following pages.

Notes

Using the MODE command

The following is a list of tasks for which you can use the MODE command. Following each task is the name of the command description as it appears in this chapter.

- Reconfiguring a printer attached to a parallel port (PRN, LPT1, LPT2, or LPT3) for printing at 80 or 132 characters per line, 6 or 8 lines per inch, or both (if the printer supports these features). See MODE (configure printer).
- Configuring the baud rate, parity, and number of data bits and stop bits of a serial communications port (COM1, COM2, COM3, and COM4) for use with a specific printer, modem, or other serial device. See MODE (configure serial port).
- Displaying the status of all devices or of a single device. See MODE (display device status).
- Redirecting printer output from a parallel port to a serial port so that the serial port becomes the system's default printer port. See MODE (redirect printing).
- Preparing devices for code-page switching. See MODE (set device code pages).
- Selecting another display adapter or changing the configuration of the current display adapter. See MODE (set display mode).
- Setting the typematic rate of the keyboard. See MODE (set typematic rate).

Requirement for ANSI.SYS and DISPLAY.SYS

The MODE program can perform some tasks, such as setting the display mode, only if you have included a DEVICE command for the ANSI.SYS device driver in your CONFIG.SYS file. You must install DISPLAY.SYS to use the MODE command (set device code pages) for code-page switching.

Adding mode commands to AUTOEXEC.BAT

Although you can type each form of the MODE command at the command prompt, you can also use MODE commands within your AUTOEXEC.BAT file to reconfigure your system automatically each time you turn on or restart your computer.

MODE (Configure Printer)

Configures a printer connected to a parallel printer port.

This version of the MODE command sets the characteristics for an IBM-compatible or Epson**-compatible printer connected to a parallel printer port (PRN, LPT1, LPT2, or LPT3).

Type

DOS, External, Network

Syntax

mode lpt*n*[:] [*c*][,[*l*][,*r*]]

mode lpt*n*[:] [**cols**=*c*] [**lines**=*l*] [**retry=***r*]

Parameters

lpt*n* Specifies the parallel port to which the device is attached. Valid values for *n* are in the range 1 through 3.

If you omit any of the following three parameters, the most recent setting for the omitted parameter is used. If you are using the shorter form of the syntax (without the words **cols=**, **lines=**, and **retry=**), the MODE command "recognizes" the parameters by their positions. Thus, if you do not specify a value for a parameter, you must still type the comma that precedes the next parameter.

cols=*c* Specifies the number of characters (columns) per line: 80 or 132. The default value is 80. You can abbreviate this parameter by simply omitting **cols=** and specifying a value for *c*.

lines=*l* Specifies the vertical spacing and the number of lines per inch: 6 or 8. The default value is 6. You can abbreviate this parameter by simply omitting **lines=** and specifying a value for *l*.

retry=*r* Specifies the retry action to take if a timeout error occurs when the MODE program attempts to send output to a parallel printer. This parameter causes part of the MODE program to remain resident in memory. The following list shows each valid value for *r* and a brief description of its meaning:

e Return an error from a status check of a busy port.

b Return "busy" from a status check of a busy port.

p Continue retrying until printer accepts output.

r Return "ready" from a status check of a busy port.

n Take no retry action (default value). You can also specify **none** for this value.

If you are using the MODE command over a network, do not use any of the *retry* values.

You can abbreviate this parameter by simply omitting **retry=** and specifying a value for *r*.

Notes

Update to mode parameter

The **retry=b** setting provides the same support as the *p* parameter did in previous versions of DOS.

Breaking out of a timeout loop

To break out of a timeout loop, press **Ctrl+C**.

Setting parallel-printer modes

For parallel-printer modes, you can use PRN and LPT1 interchangeably.

Examples

Suppose you want to be able to print 80 characters per line and 8 lines per inch on a parallel printer that is connected to the second parallel printer port (LPT2). To do this, type the following command:

```
mode lpt2:80,8
```

Because 80 characters per line is the default setting, you could achieve the same result typing the following command:

```
mode lpt2:,8
```

Suppose that, when printing a file, you want your system to keep trying to print the file until it is successful. To do this, type the following command:

```
mode lpt2:,8,b
```

To stop your system from continually retrying to print, press **Ctrl+C** or type the MODE command without specifying a value for *r*.

MODE (Configure Serial Port)

Configures a serial communications port.

This version of the MODE command sets the parameters for a serial port (COM1, COM2, COM3, or COM4).

Type

DOS, External, Network

Syntax

mode com*m*[:] [*b*[,*p*[,*d*[,*s*[,*r*]]]]]

mode com*m*[:] [**baud=***b*] [**parity=***p*] [**data=***d*] [**stop=***s*] [**retry=***r*]

Parameters

com*m* Specifies the number of the asynchronous communications (COM) port. Valid values for *m* are in the range 1 through 4.

If you omit any of the following five parameters, the most recent setting for the omitted parameter is used. If you are using the shorter form of the syntax (without the words **baud=**, **parity=**, **data=**, and so on), the MODE command "recognizes" the parameters by their positions. Thus, if you do not specify a value for a parameter, you must still type the comma that precedes the next parameter.

baud=*b* Specifies the first two digits of the transmission rate in bits per second. The following list shows each valid value for *b* and its related rate:

11	110 baud
15	150 baud
30	300 baud
60	600 baud
12	1200 baud
24	2400 baud
48	4800 baud
96	9600 baud
19	19,200 baud

The *b* value of 19 is not supported on all computers (check your hardware manual). You can abbreviate this parameter by simply omitting **baud=** and specifying a value for *b*.

parity=*p* Specifies how the system uses the parity bit to check for transmission errors. The **p** value can be one of the following: **n** (none), **e** (even), **o** (odd), **m** (mark), or **s** (space). The default value is **e**. Not all computers support the values **m** and **s**. You can abbreviate this parameter by simply omitting **parity=** and specifying a value for *p*.

data=*d* Specifies the number of data bits in a character. Valid values for *d* are in the range 5 through 8. The default value is 7. Not all computers support the values 5 and 6. You can abbreviate this parameter by simply omitting **data=** and specifying a value for *d*.

stop=*s* Specifies the number of stop bits that define the end of a character: 1, 1.5, or 2. If the baud rate is 110, the default value is 2; otherwise, the default value is 1. Not all computers support the value 1.5. You can abbreviate this parameter by simply omitting **stop=** and specifying a value for *s*.

retry=*r* Specifies the retry action to take if a timeout error occurs when the MODE program attempts to send output to a parallel printer. This parameter causes part of the MODE program to remain resident in memory. The following list shows each valid value for *r* and a brief description of its meaning:

e Return an error from a status check of a busy port.

b Return "busy" from a status check of a busy port.

p Continue retrying until printer accepts output.

r Return "ready" from a status check of a busy port.

n Take no retry action (default value). You can also specify **none** for this value.

If you are using the MODE command over a network, do not use any of the *r* values. You can abbreviate this parameter by simply omitting **retry=** and specifying a value for *r*.

Notes

The **retry=b** setting provides the same support as the *p* parameter did in previous versions of DOS.

MODE (Display Device Status)

Displays the status of one or all of the devices installed on your system.

Type

DOS, External, Network

Syntax

mode [*device*] [**/status**]

To display the status of all devices installed on your system, use the following syntax:

mode

Parameters

device Specifies the name of the device for which you want to display the status.

Switches

/status Requests the status of any redirected parallel printers. The MODE command, when used without this switch, displays the status of all installed devices except redirected parallel printers. You can abbreviate the **/status** switch as **/sta**.

Examples

To display the status of the console, type the following command:

```
    mode con
```

MODE (Redirect Printing)

Redirects output from a parallel port to a serial communications port.

Type

DOS, External, Network

Syntax

mode lpt*n*[:]**=com***m*[:]

Parameters

lpt*n* Specifies the parallel port. Valid values for *n* are in the range 1 through 3.

com*m* Specifies the serial port. Valid values for *m* are in the range 1 through 4.

Examples

Suppose you want to set up your system so that it sends parallel printer output to a serial printer. To do this, you must use the MODE command twice. The first time, you configure the serial port; the second time, you use the MODE command to redirect parallel printer output to the serial port you specified in the first MODE command.

For example, if your serial printer operates at 4800 baud with even parity and is connected to the COM1 port (the first serial connection on your computer), you would type the following two commands:

```
    mode com1 48,e,,,b

    mode lpt1=com1
```

If you redirect parallel-printer output from LPT1 to COM1 but then decide that you want to print a file by using LPT1, use the following command before you print the file. This command prevents DOS from redirecting the file from LPT1 to COM1.

```
mode lpt1
```

MODE (Set Device Code Pages)

Prepares, selects, refreshes, or displays the numbers of the code pages for parallel printers or the console.

Type

DOS, External, Network

Syntax

mode *device* **codepage prepare=((***yyy* [...]) [*drive*:][*path*]*filename***)**

mode *device* **codepage select=***yyy*

mode *device* **codepage refresh**

mode *device* **codepage** [**/status**]

Parameters

device Specifies the device for which you want to prepare or select a code page. Valid names for *device* are **con**, **lpt1**, **lpt2**, and **lpt3**.

codepage prepare
Prepares code pages for the specified device. You must prepare a code page for a device before you can use the code page with that device. After you use the **codepage prepare** form of the MODE command, use the **codepage select** form of the MODE command to specify the code page you want to use. You can abbreviate **codepage** and **prepare** as **cp** and **prep**, respectively.

yyy Specifies the number of the code page to prepare or select. The following list shows each code page that DOS supports and its country or language:

437	United States (hardware code page)
850	Multilingual (Latin I)
852	Slavic (Latin II)
855	Cyrillic
857	Turkish
860	Portuguese
861	Icelandic
863	Canadian-French
866	Russian (Russian DOS only)
869	Greek

[*drive*:][*path*]*filename*
Specifies the location and name of the code-page-information (.CPI) file that DOS uses to prepare a code page for the specified device.

codepage select
Specifies (selects) which code page to use with the specified device. Before selecting a code page, you must use the **codepage prepare** form of the MODE command to prepare a code page. You can abbreviate **codepage** and **select** as **cp** and **sel**, respectively.

codepage refresh
Reinstates the prepared code pages if they are lost as the result of a hardware problem or other error. You can abbreviate **codepage** and **refresh** as **cp** and **ref**, respectively.

codepage
Displays the numbers of the code pages, if any, that are prepared or selected for the specified device.

Switches

/status Displays the numbers of the current code pages prepared or selected for the specified device. You can abbreviate this switch as **/sta**. Whether or not you specify the **/status** switch, typing the MODE command with a device name and the **codepage** parameter displays the numbers of the code pages that are prepared or selected for the specified device.

Notes

DOS includes the following .CPI files, which correspond to specific devices:

File	**Device**
EGA.CPI	Enhanced graphics adapter (EGA) or IBM Personal System/2* (for code pages 437, 850, 852, 855, 860, and 863)
EGAX.CPI	Enhanced graphics adapter (EGA) or IBM Personal System/2 (for code pages 857, 861, and 869)
4201.CPI	IBM Proprinters II and III Model 4201, IBM Proprinters II and III XL Model 4202
4208.CPI	IBM Proprinter X24E Model 4207, IBM Proprinter XL24E Model 4208
PPDS.CPI	IBM LaserPrinter Model 4019

Related Information

For information about other code page commands, see command "NLSFUNC" on page 173 and command "CHCP" on page 31.

MODE (Set Display Mode)

Selects the active display adapter and its display mode, or reconfigures the active display adapter.

Type

DOS, External, Network

Syntax

mode [*display-adapter*][**,***shift*[**,t**]]

mode [*display-adapter*][**,***n*]

mode con[**:**] [**cols=***c*] [**lines=***n*]

Parameters

display-adapter

Specifies a setting category. The following list shows the values associated with each setting category for *display-adapter*.

40 or **80**

Indicates the number of characters per line.

bw40 or **bw80**

Specifies a color graphics adapter (CGA) with color disabled, and specifies the number of characters per line.

co40 or **co80**

Specifies a color monitor with color enabled, and specifies the number of characters per line.

mono Specifies a monochrome display adapter with a constant width of 80 characters per line.

shift Specifies whether to shift the CGA screen to the left or to the right. Valid values for *shift* are **l** (for left) and **r** (for right).

t Enables you to align the screen by using a test pattern. DOS prompts you to indicate whether the screen is aligned correctly.

con[:] Refers to the monitor.

cols=*c* Specifies the number of characters (columns) per line. Valid values are 40 and 80.

lines=*n*

Specifies the number of lines that can be displayed on the screen. Valid values for *n* are 25, 43, and 50. Not all display adapters support all three settings. To set the number of lines, you must have installed the ANSI.SYS device driver by using a DEVICE command in your CONFIG.SYS file.

MODE (Set Typematic Rate)

Sets the keyboard typematic rate, the rate at which DOS repeats a character when you hold down the key for that character.

The typematic rate has two components, the rate and the delay. Some keyboards do not recognize this command.

Type

DOS, External, Network

Syntax

mode con[:] [**rate=***r* **delay=***d*]

Parameters

con[:] Refers to the keyboard.

rate=*r* Specifies the rate at which a character is repeated on the screen when you hold down a key. Valid values are in the range 1 through 32. These values are equal to approximately 2 to 30 characters per second, respectively. The default value is 20 for IBM AT-compatible keyboards, and 21 for IBM PS/2-compatible keyboards. If you set the rate, you must also set the delay.

delay=*d*
Specifies the amount of time that must elapse—after you press and hold down a key—before DOS starts to repeat the character. Valid values for *d* are 1, 2, 3, and 4 (representing 0.25, 0.50, 0.75, and 1 second, respectively). The default value is 2. If you set the delay, you must also set the rate.

MORE

Displays one screen of output at a time.

The MORE command reads standard input from a pipe or redirected file and displays one screen of information at a time. This command is commonly used to view long files.

Type

DOS, External, Network

Syntax

more < [*drive*:][*path*]*filename*

command-name | **more**

Parameters

[*drive*:][*path*]*filename*
Specifies the location and name of a file that supplies data you want to display.

command-name
Specifies the command that supplies data you want to display.

Notes

Sources of data

When using the redirection character (<), you must specify a file name as the source. When using the pipe (|), you can use such commands as DIR, SORT, and TYPE. Before using a pipe for redirection, you should set the TEMP environment variable in your AUTOEXEC.BAT file.

Examples

Suppose you have a long file named CLIENTS.NEW that you want to view on your screen. Either of the following two commands redirects the file through the MORE command to begin displaying the contents of the file:

```
more < clients.new

type clients.new | more
```

The MORE command displays the first screen of information from CLIENTS.NEW and then prompts you with the following message:

```
- More -
```

You can then press any key to see the next screen of information.

Related Information

For information about displaying the contents of a directory, see command "DIR" on page 72.

For information about displaying the contents of a file, see command "TYPE" on page 230.

MOUSE

Enables DOS to detect if a mouse is connected to your computer and to copy MOUSE.COM to your DOS directory. Then, a path statement that leads to the MOUSE.COM file is added to the AUTOEXEC.BAT file.

Type

DOS

Syntax

mouse [*drive:*][*path*][/**b**[/**c***n*][/**l***n*][/**z**] [/**r***n*][/**h***n*][/**s***n*][/**v***n*][/**m**][/**n***n*][/**p***n*][/**y**] [/**O***r*][/**kc**][/**kp***nsm*][/**kp***nsm*]*off*

Parameters

drive Specifies the drive that contains the MOUSE.COM file.

/path
Specifies the path leading to the location where the MOUSE.COM file is stored.

off Disables mouse support.

Switches

/b Specifies a Bus mouse type.

/c*n* Specifies a serial mouse type. *n* can have a value of 1 or 2.

/i*n* Specifies an inPort mouse type. *n* can have a value of 1 or 2.

/z Specifies a PS2 mouse type.

/r*n* Specifies the interrupt rate. The value for *n* is 0-4.

/h*n* Specifies the horizontal sensitivity of mouse movements. The horizontal value for *n* is 0 - 100.

/s*n* Specifies the horizontal and vertical sensitivity of mouse movements. The *n* represents horizontal and vertical ranges from 0 - 100 only.

/v*n* Specifies the vertical sensitivity of mouse movements. The vertical value for *n* is 0 - 100.

/m Forces the default cursor to on.

/n*n* Specifies the cursor display delay. The value for *n* is 0 - 255.

/p*n* Specifies the active acceleration profile. The value for *n* is 1 - 4.

/y Specifies that the hardware cursor be turned off.

/or Specifies the rotation angle. The value for *n* is 0 - 359.

/kc Specifies that the click lock be turned on.

/kp*nsm*
Specifies the primary button selection.

/kp*nsm*
Specifies the secondary button selection.

off Disables mouse support from your system.

Notes

Interrupt rate ranges The following values are used if you have a serial mouse:

0=	0Hz
1=	30Hz

2=	50Hz
3=	100Hz
4	200Hz

The following values are used if you have a PS2 mouse:

0=	20Hz
1=	40Hz
2=	60Hz
3=	80Hz
4=	100Hz

Improving pointer visibility on a liquid crystal display

To improve mouse pointer visibility on computers with liquid crystal (LCD) displays, this version of the mouse software introduces several new features. Pointer visibility is improved by altering the size and color of the pointer. The pointer size can be set up to become enlarged while the mouse is moving, returning to its normal size when mouse movement slows. This lets you keep track of the pointer while it is in motion and you continue to do detailed work with the pointer at its normal size. Pointer settings are found in the MOUSE.INI file.

In DOS text mode, the normal pointer has the shape of a square box. To improve the visibility on LCD screens in text mode, the medium pointer has the shape of a cross and the large pointer has the shape of a large box with a hole in the middle.

Note: In DOS graphics mode, specifying medium or large does not change the arrow pointer size. Instead, the pointer is surrounded by a rounded box to improve visibility.

Pointer settings are defined in the MOUSE.INI file under **PointerSize** and **PointerColor**. The MOUSE.INI file is located in the specified DOS directory. For example:

```
C:\DOS\MOUSE
```

For additional information about the MOUSE.INI file and how you can edit it, see Chapter 6, “Configuring Mouse and RAMBoost .INI Files” on page 289.

Examples

The following examples show how to load and disable the mouse in your system.

To load the mouse driver into memory, type the following at the DOS command prompt.

```
mouse
```

To disable mouse support, type the following at the DOS command prompt.

```
mouse off
```

MOVE

Moves files and renames files and directories.

Type

DOS, External

Syntax

To move one or more files:

move [/y][*drive*:] [*path*]*filename*[[*drive*:][*path*]*filename*[..]]*destination*

To rename a directory:

move [*drive*:][*path*]*dirname1 dirname2*

Parameters

[*drive*:][*path*]*filename*
Specifies the location and name of the file or files you want to move.

destination
Specifies the new location of the file. Destination can consist of a drive letter and colon, a directory name, or a combination. If you are moving only one file, you can also include a file name if you want to rename the file when you move it.

[*drive*:][*path*]*dirname1*
Specifies the directory you want to rename.

dirname2
Specifies the new name of the directory.

Switches

/y
Specifies that prompting be suppressed if a target directory needs to be created. The prompt only occurs if you are moving more than one file to a target directory that does not exist.

Notes

Destination is an existing file

If the destination you specify is an existing file, MOVE replaces that file with the file you are moving.

Moving more than one file at a time

You can move more than one file at a time to the same destination. Ensure that you separate the filespecs. For example, to move the files TEXT.DOC and MEMO.DOC to the LETTERS directory, type the following:

```
    move text.doc,memo.doc letters
```

When you move more than one file at a time, the destination must include a directory name.

NLSFUNC

Starts the Nlsfunc program, which loads country-specific information for national language support (NLS).

You can use the NLSFUNC command either from the command line or within CONFIG.SYS to support the use of country-specific information and code-page switching.

Type

DOS, External, Network

Syntax

nlsfunc [[*drive*:][*path*]*filename*]

In your CONFIG.SYS file, use the following syntax:

install=[[*dos-drive*:]*dos-path*]**nlsfunc.exe** [*country-filename*]

Parameters

[*drive*:][*path*]*filename* or *country-filename*

Specifies the location and name of the file containing country-specific information. If you use this parameter in the INSTALL command, you must include the drive and directory.

[*dos-drive*:]*dos-path*

Specifies the location of NLSFUNC.EXE.

Notes

The default value for [*drive*:][*path*]*filename* is defined by the COUNTRY command in your CONFIG.SYS file. If no COUNTRY command exists in the CONFIG.SYS file, the Nlsfunc program looks for the COUNTRY.SYS file in the root directory of the startup drive. The Nlsfunc program does not access the COUNTRY.SYS file until DOS requests information from it. If DOS cannot find the COUNTRY.SYS file when you install the Nlsfunc program, no error message is given. However, you will get an error message if you subsequently run a CHCP command.

Examples

To use the default country-specific information found in the COUNTRY.SYS file, type the following command:

```
nlsfunc
```

Suppose you have a file called NEWCDPG.SYS that contains country-specific information. If you want to use the information from that file rather than the COUNTRY.SYS file, type the following command:

```
nlsfunc newcdpg.sys
```

Related Information

For information about displaying the current code page, see command “CHCP” on page 31.

For information about preparing a code page, see command “MODE” on page 158.

NUMLOCK

Specifies whether the Num Lock key is set to ON or OFF when your computer starts. You can use this command only in your CONFIG.SYS file.

Type

CONFIG.SYS

Syntax

numlock=[on|off]

Parameters

on|off On specifies that the **Num Lock** key be set to *on* when DOS displays the startup menu. Off specifies that the **Num Lock** key be set to *off* when DOS displays the startup menu.

Notes

Ensuring that the Num Lock key is always on

To ensure that the Num Lock key is always on, include the NUMLOCK command within the [MENU&brbk. block in your CONFIG.SYS file.

Related Information

For information about the [MENU] command, see command “[MENU]” on page 151.

PATH

Sets a search path for executable files.

DOS uses the PATH command to search for executable files in the directories you specify. By default, the search path is the current directory only.

Type

DOS, Internal, Network

Syntax

path [[*drive*:]*path*[;...]]

To display the current search path, use the following syntax:

path

To clear all search-path settings other than the default setting (the current directory), use the following syntax:

path ;

Parameters

[*drive*:]*path*
Specifies a drive, directory, and any subdirectories to search.

; When used as the only parameter, clears all search-path settings and specifies that DOS is to search only the current directory.

Notes

Current directory

DOS always searches in the current directory first, before it searches directories in the search path.

Length limit for the PATH command

The maximum length of the PATH command is 127 characters. To fit more directories in the search path, you can shorten directory names, use the SUBST command to redirect directories to logical drives (which shortens the entries on the PATH command line), or use the APPEND **/x:on** command.

Files with the same name, different extensions

You might have some files in the same directory that share the same file name but have different extensions. For example, you might have a file named ACCNT.COM that starts an accounting program and another file named ACCNT.BAT that connects your system to the accounting system network.

DOS searches for a file by using default file name extensions in the following order of precedence: .COM, .EXE, and .BAT. To run ACCNT.BAT when ACCNT.COM exists in the same directory, you must include the .BAT extension on the command line.

Two or more identical file names in the path

You might have two or more files in the search path that have the same file name and extension. DOS searches for the specified file name first in the current directory. Then it searches directories in the order in which they are listed in the PATH command.

Specifying multiple paths

To specify more than one path for DOS to search, separate entries with a semicolon (;).

Using path in your AUTOEXEC.BAT file

If you place the PATH command in your AUTOEXEC.BAT file, DOS automatically initiates the specified search path every time you start your computer.

Examples

The following command specifies that DOS is to search three directories to find external commands (the three paths for these directories are C:\USER\TAXES, B:\USER\INVEST, and B:\BIN):

```
path c:\user\taxes;b:\user\invest;b:\bin
```

Related Information

For information about setting a search path for data files, see command "APPEND" on page 21.

PAUSE

Suspends processing of a batch program and displays a message prompting the user to press any key to continue.

Type

Batch, Internal

Syntax

pause

Notes

Prompting the user to continue the program

DOS displays the following message in response to the PAUSE command:

```
Press any key to continue...
```

Dividing a batch file into sections

If you press **Ctrl+C** to stop a batch program, DOS displays the following message:

```
Terminate batch job (Y/N)?
```

If you press **Y** (for yes) in response to this message, the batch program ends and control returns to the operating system. Therefore, you can insert the PAUSE command before a section of the batch file you do not want to process. While PAUSE suspends processing of the batch program, you can press **Ctrl+C** and then **Y** to stop the batch program.

Examples

Suppose you want a batch program to prompt the user to change diskettes in one of the drives. To do this, you might create the following file:

```
@echo off
:begin
copy a:*.*
echo Please put a new diskette into drive A
pause
goto begin
```

In this example, all the files on the diskette in drive A are copied to the current directory. After the displayed comment prompts you to place another diskette in drive A, the PAUSE command suspends processing so that you can change diskettes and then press any key to resume processing. This particular batch program runs in an endless loop. The GOTO *begin* command sends the command interpreter to the *begin* label of the batch file. To stop this batch program, press **Ctrl+C** and then **Y**.

PCFORMAT

Formats a PCMCIA static random access memory (SRAM) card to any size for DOS File Allocation Table (FAT) file system drivers.

Type

DOS, External

Syntax

pcformat *drive*: [**/s**:*nnnn* k | m][/**?**]

Parameters

drive: Specifies the drive letter containing the SRAM card to be formatted.

Switches

/s:*nnnn* k | m
Specifies the card size (*nnnn*) in Kilobytes (K) or Megabytes (M). The maximum size supported is 32 MB. The K or M qualier is required.

/? Displays help messages about command line parameters and their associated syntax.

PCMATA.SYS

This is an installable device driver provided by DOS. See "PCMATA.SYS" on page 268 for more details.

PCMCS

PCMCS is the command that starts the PCMCS.EXE Card Services 2.0 driver that interfaces directly with Socket services 2.0 which meets the Intel Exchangeable Card Architecture (ExCA).

PCMCS can be loaded as a terminate-and-stay-resident program from either the DOS prompt, the AUTOEXEC.BAT file, or as a device driver from the CONFIG.SYS file.

Type

DOS, External

Syntax

When loaded from the DOS prompt or the AUTOEXEC.BAT file, use the following syntax:

[*drive*:][*path*] **pcmcs**[**/wait=***n*][**/addr=***xx*] [**/irq=***n*][**/clients=***n*] [**/pmoff**] [**/regions**][/**?**]

Note: Socket Services is not installed with DOS. This software is provided with your system. If an error occurs while attempting to load Card Services, refer to the README.TXT file in your DOS directory for information about Socket Services.

For information about how to load as a device driver in your CONFIG.SYS file, see "PCMCS.EXE" on page 270.

Parameters

[*drive*:][*path*]
Specifies the location of the device driver file.

Switches

/wait=*n* Specifies the PCMCIA card insert settle time. This time is the delay time needed from the time a card is inserted in a socket to the time that Card Services can access it.

The value of *n* is the number of system timer ticks (18.2 per second) to wait. The default is 12. Some cards require a longer settle time than others.

/addr=*xx*
Specifies the starting segment address for Card Services for PCMCIA card configuration.

The value for *xx* is Start address. You must locate the segment (Start Address) within the first 1MB of address space and must specify the start address of a 2-digit hexadecimal segment address.

The granularity of the start address is dependent on the socket controller. However, the minimum address granularity allowed is 4K. If *addr* is not specified, PCMCS defaults to the first available 4K block starting at hex C0. The range for *xx* is hex C0 to F0 (where C0 is equivalent to hex C000, and hex F0 is equivalent to hex F000).

/irq=*n* Specifies the IRQ resource that is used by Card Services for PCMCIA card events. The value of *n* must be in the range of 8 to 15. If this switch is not specified, the default is 10.

/clients=*n*
Controls the maximum number of client drivers that can be registered with Card Services. Each client driver requires Card Services to allocate 60 bytes of memory. If this switch is not specified, the default is 10.

/pmoff This option controls the power management provided with Card Services. If this option is specified, PCMCIA cards are unilaterally powered-down on SUSPEND messages, and on RESUME messages the cards are powered-up. The cards are also reconfigured (if there is an associated client driver) by sending an artificial insertion message. When the system resumes, the cards must be reconfigured.

/REGIONS=*n*
Defines the maximum number of concurrent regions that Card Services will manage. A *region* corresponds to MTDs and is added to reduce the amount of memory required for Card Services.

/? This option displays a help message regarding command line parameters and their syntax.

Notes

Determining the default Resource Map

Card Services is responsible for building a map of resources available to PCMCIA cards. When PCMCS starts, all standard devices are examined and the resources for these are devices reserved to avoid conflicting with each other.

The variables are for reserving resources used by other devices that your computer might have (I/O, memory, and IRQs) that are not considered standard in an AT compatible system.

Card Services, when started, examines the following:

- COM devices (COM1 - COM4)
- LPT devices (LPT1 - LPT3)
- Hard drive controllers (primary and secondary)
- Diskette drive controllers (primary and secondary)
- VGA adapters
- EGA adapters
- CGA adapter
- Monochrome adapter
- Game adapter (joystick)

Related Information

For more information about the PCMCS device driver, see “PCMCS.EXE” on page 270.

For information about how the PCMCIA program works, see *PC DOS User's Guide*.

PCMCS.EXE

This is an installable device driver provided by DOS. See “PCMCS.EXE” on page 270 for more detail.

PCMFDD

Provides diskette drive emulation on PCMCIA sockets as drive A and B.

PCMFDD can be loaded either as a terminate-and-stay-resident program from either the DOS prompt, the AUTOEXEC.BAT file, or as a device driver from the CONFIG.SYS file.

Type

DOS, External

Syntax

When loaded from the DOS prompt or the AUTOEXEC.BAT file, use the following syntax:

[*drive*:][*path*] **pcmfdd.exe** /addr=*nn* /*x*:*m*

When loaded from the CONFIG.SYS file, include the following statement in your CONFIG.SYS file:

device=[*drive*:][*path*] **pcmfdd.exe** /addr=*nn* /*x*:*m*

Parameters

[*drive*:][*path*]
Specifies the location of the device driver file.

Switches

/addr=*nn* Specifies the system window base address of 8K for memory card emulation. The base address must be on a 16K boundary, where *nn* ranges from C0 to EE (C0 is equivalent to C000H, and EE is equivalent to EE00H)

x The emulated drive letter (drive A or drive B).

m The socket number (0 or 1).

Related Information

For more information about the PCMFDD.EXE device driver, see "PCMFDD.EXE" on page 271.

PCMFDD.EXE

This is an installable device driver provided by DOS. See "PCMFDD.EXE" on page 271 for more information.

PCMINFO

Provides information about each PCMCIA socket and the PC Card inserted in it. Information that is displayed, for example, includes: the card, name, product name, device type, PCMCIA Services version, and Card Services vendor version number.

Type

DOS, External

Syntax

pcminfo [/d] [/t][/?]

Switches

/d Continuously monitors the sockets, displaying the results.

/t Suppresses timer tick display.

/? Displays help information.

Notes

Windows (version 3.1) users

During the install of this version of DOS, if you choose PCMCIA Support, the WPCMINFO.CPL and WPCMINFO.HLP files are copied into your DOS directory. If you have Windows (version 3.1) or install Windows after installing this version of DOS, you need to copy these two files into your WINDOWS\SYSTEM directory. A PCMCIA icon is created in the Control Panel under Windows. Clicking on this icon accesses the PCMCIA support information utility.

PCMMTD

PCMMTD provides SHELL.MTD, which demonstrates all of the interface capabilities in the MTD.

PCMMTD.EXE can be loaded either as a terminate-and-stay resident program from the DOS command prompt or as a device driver from the CONFIG.SYS file. The PCMCS.EXE must be loaded prior to loading PCMMTD.EXE.

Syntax

Add the following device driver to your CONFIG.SYS file to install the Memory Technology Driver:

```
device=[drive:][path]pcmmtd.exe
```

From the DOS command prompt, type:

```
[drive:][path]pcmmtd.exe
```

Parameters

[*drive:*] [*path*]	Specifies the location of the device driver.

Related Information

For information about the PCMMTD.EXE device driver, see "PCMMTD.EXE" on page 272.

PCMMTD.EXE

This is one of the installable device drivers provided with DOS. See "PCMMTD.EXE" on page 272 for more information.

PCMSCD

The PCMSCD.EXE is a super client driver that supports the configuration of several PC Cards. After configuration, the PC Card operates as an integral component of the system.

The PCMCIA Super Client Driver is intended to work only with the Phoenix PCMCIA Card Services and does not operate on Card Services furnished by third party suppliers.

You can either load the Super Client Driver as a device driver from the CONFIG.SYS file, run from the DOS command promp (loading it as a TSR), or add this command to your AUTOEXEC.BAT file.

Syntax

When loaded from the DOS prompt or the AUTOEXEC.BAT file, use the following syntax:

[*drive*:][*path*] **pcmscd** [**/beep**][**/rs**=*speed*][**/com**=*port*][**/cards**][**/?**]

When loaded from the CONFIG.SYS file, include the following statement in your CONFIG.SYS file:

device=[*drive*:][*path*]**pcmscd.exe** [**/beep**][**/rs**=*speed*][**/com**=*port*][**/cards**][**/?**]

Parameters

[*drive*:][*path*] Specifies the location of the device driver file.

Switches

/beep Audible configuration acknowledgement. The default value is for the beep to be turned off.

/com=*port* Number of the communication port, where *port* can be 1, 2, 3, or 4. The default is 4.

/rs=*speed* IBM Token-Ring Network speed, where *speed* is 4 or 16. The default value is 4.

/? Displays help information.

/cards Displays a list of supported cards.

PCMSCD.EXE

This is one of the installable device drivers provided with DOS. See "PCMSCD.EXE" on page 273 for more information.

PCMVCD.386

This is one of the installable device drivers provided by DOS. See "PCMVCD.386" on page 274 for more information.

PENDOS

The PENDOS command starts the PenDOS** program.

Type

DOS, External

Syntax

pendos

Notes

The following support programs are all part of the PenDOS program.

- VLOAD
- PKEYUS
- PMOUSE
- PINK
- PEN
- PSYS
- PWW

These internal programs do not require user interface. They are described for the purpose of providing an insight into how the PenDos program works.

For more information about how the PenDOS program works, see the *PC DOS User's Guide*.

VLOAD

The VLOAD program loads the PENDOS handwriting recognition system into expanded memory provided by the EMM386 extended memory manager.

The syntax for this program is:

```
VLOAD [filename]
```

filename Specifies the handwriting recognition engine to be loaded.

Note: EMM386.EXE must be loaded in order to use the VLOAD command, and at least 256K of expanded memory must be available.

PKEYUS

The PKEYUS command starts the PKEYUS program which provides PenDOS virtual keyboard support for pen-based systems.

The PKEYUS program is a terminate-and-stay-resident program.

The syntax for this program is:

```
PKEYUS
```

Note: The PKEYUS program is the US keyboard version. Other language keyboard layouts will have different file names (for example, PKEYFR would be used for French).

PMOUSE

Starts the PMOUSE program which provides PenDOS mouse emulation for pen-based systems. The PMOUSE program is a terminate-and-stay-resident program.

The syntax for this program is:

```
PMOUSE
```

PINK

The PINK program provides support for electronic ink for PenDOS (which enables you to produce script with the pen device).

The PINK program is a terminate-and-stay-resident program.

The syntax for this program is:

```
PINK
```

PEN

The PEN command starts the PEN program which provides PenDOS point processing routines, interface to digitizer driver, and display routines for pen-based systems.

The PEN program is a terminate-and-stay-resident program.

The syntax for this program is:

```
PEN
```

PSYS

The PSYS command starts the PSYS program which provides the PenDOS program the initial system buttons for the buffered keyboard.

The PSYS program is a terminate-and-stay-resident program.

The syntax for this program is:

```
PSYS
```

PWW

The PWW command starts the PWW program which provides support for the PenDOS Writing Window.

The PWW program is a terminate-and-stay-resident program.

The syntax for this program is:

```
PWW b
```

PENDEV.SYS

This is one of the installable device drivers provided by DOS. It provides the PenDOS application programming interface (API) for PenDOS applications.

Related Information

For detailed information about this device driver, see “PENDEV.SYS” on page 276.

POWER

Starts the Advanced Power Management (APM) program that provides the ability to reduce the consumption of power when your applications and devices are idle.

Type

DOS, External

Syntax

power [adv[:*max*|*reg*|*min*]|**std|off]**

Switches

[adv[:*max*|*reg*|*min* **]**
Conserves power when applications and hardware devices are idle. Performance might be affected if an application is active instead of idle. Use *max* for maximum power conservation. Use *reg*, the default setting, to balance power conservation with application and device performance. Use *min* if the performance of an application or device is not satisfactory when you specify *max* or *reg*.

std
Conserves power by using only the power-management features of your computer's hardware.

off
Turns off power management.

Notes

A command-line statement must be in your CONFIG.SYS file before POWER can work. It must specify the location of the POWER.EXE file. For example:

```
device=c:\dos\power.exe
```

Examples

After the device driver is loaded, POWER.EXE can run from the DOS command line or BATCH file.

To display the current power setting, type the following:

```
power
```

To change the current power setting, type the following:

```
power adv:max
```

This changes the current power setting for maximum power conservation.

POWER.EXE

This is one of the installable device drivers provided by DOS. It provides the ability to reduce the consumption of power when your applications and devices are idle.

Related Information

For detailed information about this device driver, see "POWER.EXE" on page 276.

PRINT

Prints a text file while you are using other DOS commands.

This command can print in the background if you have an output device connected to one of the serial or parallel ports of your system.

Type

DOS, External, Network

Syntax

print [/**d**:*device*][/**b**:*size*][/**u**:*ticks1*][/**m**:*ticks2*] [/**s**:*ticks3*] [/**q**:*qsize*][/**t**][[*drive*:][*path*]*filename*[...]] [/**c**] [/**p**]

To install the Print program with the default parameters or to display the contents of the print queue on your screen without affecting the queue, do the following:

1. Type `print` at the command prompt. A message is displayed requesting the device name.
2. Type the device name and press **Enter** A message is displayed indicating that the Print program has been installed and if there are any jobs waiting in the print queue these are listed.

Parameters

[*drive*:][*path*]*filename*
: Specifies the location and name of a file or set of files you want to print. You can include multiple files (usually as many as 13) on one command line.

Switches

/**d**:*device*
: Specifies the name of the print device. Valid values for parallel ports are *lpt1*, *lpt2*, and *lpt3*. Valid values for serial ports are *com1*, *com2*, *com3*, and *com4*. The default value is *prn*. The values *prn* and *lpt1* refer to the same parallel port. The /**d** switch must precede any file name used on the command line.

/**b**:*size* Sets the size (in bytes) of the internal buffer, which is used to store data before it is sent to the printer. The minimum and default value for *size* is 512; the maximum value is 16384. Increasing this value decreases the amount of memory available for other purposes but may speed up the PRINT command.

/**u**:*ticks1*

Specifies the maximum number of clock ticks the Print program is to wait for a printer to be available (clock ticks occur about 18 times per second). If the printer is not available within the time specified, the job does not print. Values for *ticks1* must be in the range 1 through 255. The default value is 1.

/**m**:*ticks2*

Specifies the maximum number of clock ticks the Print program can take to print a character on the printer. Values for *ticks2* must be in the range 1 through 255. The default value is 2. If a character is printed too slowly, DOS displays an error message.

/**s**:*ticks3*

Specifies the number of clock ticks the DOS scheduler allocates for background printing. Values for *ticks3* must be in the range 1 through 255. The default value is 8. Increasing this value can speed up printing while slowing down other programs.

/**q**:*qsize*

Specifies the maximum number of files allowed in the print queue. Values for *qsize* must be in the range 4 through 32. The default value is 10.

/**t** Removes all files from the print queue.

/**c** Removes files from the print queue. You can use the /**c** and /**p** switches on the same command line.

When the /**c** switch precedes the list of file names on the command line, it applies to all files whose names follow the /**c** switch, until the Print program encounters a /**p** switch, in which case the /**p** switch applies to the file whose name precedes the /**p** switch.

When the /**c** switch follows a file name, it applies to the file whose name precedes the /**c** switch and all files whose names follow the /**c** switch, until the Pause program encounters a /**p** switch, in which case the /**p** switch applies to the file whose name precedes the /**p** switch.

/**p** Adds files to the print queue. You can use the /**c** and /**p** switches on the same command line.

When the /**p** switch precedes the list of file names on the command line, it applies to all files whose names follow the /**p** switch, until the Print program encounters a /**c** switch, in which case the /**c** switch applies to the file whose name precedes the /**c** switch.

When the /**p** switch follows a file name, it applies to the file whose name precedes the /**p** switch and all files whose names follow the /**p** switch, until the Print program encounters a /**c** switch, in which case the /**c** switch applies to the file whose name precedes the /**c** switch.

Notes

Length of a print queue entry

Each print queue entry can contain a maximum of 64 characters. Each queue entry includes the drive letter, directory, and any subdirectories.

Limitations on switches

You can only use the **/d**, **/b**, **/u**, **/m**, **/s**, and **/q** switches the first time you use the PRINT command after starting DOS. To use one of these switches after using the PRINT command, you need to restart DOS.

Printing files generated by programs

Many programs have their own print commands. You should use the print command of the program used in creating files to print the files.

Examples

The following command sets up the print queue for printing on LPT1:

```
print /d:lpt1
```

The following command removes the PENCIL.TST file from the print queue:

```
print pencil.tst /c
```

The next command shows how to remove the file PENCIL.TST from the queue and add the file PEN.TST to the queue:

```
print pencil.tst /c pen.tst /p
```

The following three examples use switches that work only with the first PRINT command you use after starting DOS.

To specify that the PRINT command is to wait 60 clock ticks for a printer to be available and that the DOS scheduler is to allocate 25 clock ticks to the PRINT command for background printing rather than the default value of 8 clock ticks, type the following command:

```
print /u:60 /s:25
```

The following example specifies that the Print program has 4 clock ticks available to print each character rather than the default value of 2 clock ticks:

```
print /m:4
```

To change the default maximum number of files for the print queue, use the PRINT command with the **/q** switch, as the following example shows:

```
print /q:32
```

Related Information

For information about configuring a printer connected to a parallel port, see command "MODE" on page 158.

For information about displaying the status of a printer, see command "MODE" on page 158.

For information about configuring a printer connected to a serial port, see command "MODE" on page 158.

For information about preparing printers for code-page switching, see command "MODE" on page 158.

PRINT.SYS

This is one of the installable device drivers provided by DOS. It supports code page switching for printers.

Related Information

For detailed information about this device driver, see "PRINTER.SYS" on page 277.

PROMPT

Changes the DOS command prompt.

You can customize the command prompt to display any text you want, including such information as the name of the current directory, the time and date, and the DOS version number.

Type

DOS, Internal, Network

Syntax

prompt [*text*]

Parameters

text Specifies any text and information you want included in your system prompt.

The following list shows the character combinations you can include instead of, or in addition to, any character string(s) in the *text* parameter. The list includes a brief description of the text or information that each character combination adds to your command prompt.

$q	= (equal sign)
$$	$ (dollar sign)
$t	Current time
$d	Current date
$p	Current drive and path
$v	DOS version number
$n	Current drive
$g	> (greater-than sign)
$l	< (less-than sign)
$b	| (pipe)
$_	ENTER-LINEFEED
$e	ASCII escape code (code 27)
$h	Backspace (to delete a character that has been written to the PROMPT command line)

Notes

Using the prompt command without the *text* parameter

When you use the PROMPT command without specifying a value for *text*, the command prompt is reset to the default setting, the current drive letter followed by a greater-than sign (>).

Using the $p value for *text*

If you include the *$p* character in the *text* parameter, DOS reads your disk after you enter each command to determine the current drive and path. This can take extra time, especially for diskette drives.

Examples

The following example sets the command prompt to display the current drive and path followed by the greater-than sign (>):

```
prompt $p$g
```

The following command displays a two-line prompt in which the current time appears on the first line and the current date appears on the second line:

```
prompt time is: $t$_date is: $d
```

If your CONFIG.SYS file loads ANSI.SYS, you can use ANSI escape sequences in your prompts.

The following command, for example, displays your prompt in reverse video mode and returns to usual video mode for other text:

```
prompt $e[7m$n:$e[m
```

The characters following the escape code (**$e**) are ANSI escape sequences. For information about ANSI.SYS and ANSI escape sequences, see Chapter 5, "Working With Device Drivers" on page 245.

Related Information

For information about setting the current date and time, see command "DATE" on page 61 and command "TIME" on page 228.

For information about adding the PROMPT command to your AUTOEXEC.BAT see the *PC DOS User's Guide*.

QCONFIG

The QCONFIG command displays detailed technical information about your computer.

Type

DOS, External

Syntax

qconfig[/a][/d][/o[*file*]][/p][/q][/?]

Switches

/a Use this switch to list all the Micro Channel* adapters supported by the QConfig program.

/d Use this switch to generate a detailed listing of hardware.

/o*filename* Use this switch to redirect output. If you specify a filename, the output from running the QConfig program is redirected to that file. If you do not specify a filename, the output is redirected to a file named QCONFIG.OUT.

/p Use this switch to pause the output between screens.

/q Use this switch to indicate that the redirected message is not to be displayed.

/? Provides on line help for the QCONFIG command.

Notes

Using /o*filename* with the /q switch

You can use the /**o***filename* and /**q** switches on the same command line.

RAMBOOST.EXE

This is an installable device driver provided by DOS. It increases your computer's conventional memory and makes using EMM386.EXE less complex. It also automatically analyzes your computer's current configuration and then reconfigures it to load terminate-and-stay-resident programs and device drivers into upper memory blocks.

Related Information

For detailed information about this device driver, see "RAMBOOST.EXE" on page 278.

RAMDRIVE.SYS

This is an installable device driver provided by DOS. It simulates a hard disk drive by creating a virtual disk drive in your system's random access memory (RAM).

Related Information

For detailed information about this device driver, see "RAMDRIVE.SYS" on page 280.

RAMSETUP

Starts the RAMBoost setup program. The RAMBoost setup program installs the RAMBOOST device driver in your CONFIG.SYS file and then reboots your system. The RAMBoost setup program has an advanced setup screen that you can use to customize the way RAMBOOST uses the upper memory blocks (UMB) of your computer. RAMBOOST is a full screen application.

The RAMBoost setup program modifies the RAMBOOST.INI file where system information is kept.

Type

DOS, External

Syntax

ramsetup [/25][/28][/43][/50][/60][/in][/bw][/mono][/lcd][/ff][/bf][/nf][/bt][/ngm][/le][/im][/ps2]

Switches

/25 Set screen display to 25 lines (the default).

/28 Set screen display to 28 lines (VGA only).

/43 Set screen display to 43 lines (VGA and EGA).

/50 Set screen display to 50 lines (VGA only).

/60 Set screen display to 60 lines (Video 7** only).

/in Run program in color mode even if color display device is not detected.

/bw Use black and white color scheme.

/mono Use monochrome color scheme (IBM monochrome).

/lcd Use LCD color scheme (for laptops).

/ff Speed up display (CGA only). Can cause "snow" effect on some display devices.

/bf BIOS font (use if graphics do not display properly)

/nf No fonts (do not use graphics characters).

/bt Allows graphics mouse in Windows, and graphics fonts with Desqview or UltraVision.

/ngm No graphics mouse pointer (use text character mouse pointer).

/le Left-handed mouse (exchange left and right mouse buttons).

/im Disable mouse.

/ps2 Reset mouse hardware. (Use this switch if your mouse disappears from the screen or cannot move.)

Notes

Affect on LOADHIGH and DEVICEHIGH statements

The RAMBoost program reconfigures all LOADHIGH, and DEVICEHIGH statements in your CONFIG.SYS and AUTOEXEC.BAT files. It is designed to analyze the current use of memory and then optimize the system's memory.

Related Information

For more information about the RAMBOOST.INI file see Chapter 6, "Configuring Mouse and RAMBoost .INI Files" on page 289.

For more information about the RAMBOOST.EXE see Chapter 5, "Working With Device Drivers" on page 245

For more information about using the RAMBoost program, see the *PC DOS User's Guide*.

RECOVER

Recovers readable information from a bad or defective disk.

The Recover program reads a file sector by sector and recovers data from the good sectors. Data in bad sectors is lost. All recovered data is restored to the root directory.

CAUTION:
Since the root directory can hold only a finite number of entries, some files may be lost. If you need to recover all files on a disk, recover them one at a time. Do not attempt to recover all the files in a directory or on a disk at one time unless absolutely necessary.

Type

DOS, External

Syntax

recover [*drive*:][*path*]*filename*

To recover all files on a disk when the disk directory is unusable, use the following syntax:

recover *drive*:

Parameters

[*drive*:][*path*]*filename*
Specifies the location and name of the file you want to recover. (Use this syntax to recover a single file.)

drive:
Specifies the drive from which you want to recover all files.

Notes

Limitation on [*drive*:][*path*]*filename*

You cannot use wildcards (* and ?) with the RECOVER command. You must specify either a file or a drive.

Locating recovered files

When you recover an entire disk, each recovered file is placed in the root directory in a FILE*nnnn*.REC file, where *nnnn* is a 4-digit number. The first recovered file is named FILE0001.REC, the next recovered file is named FILE0002.REC, and so on.

Reentering lost data

Since all data in bad sectors is lost when you recover a file, you should recover files one at a time. This method allows you to edit each file and reenter missing information after you recover the file.

Recovering bad sectors

Bad sectors reported by using the CHKDSK command are marked as "bad" when your disk is first prepared for operation. They pose no danger, and the RECOVER command does not affect them.

Limitations with networks and assigned drives

The RECOVER command cannot recover files on a network drive. Also, the Recover program does not work on a drive formed by the ASSIGN, JOIN, or SUBST command.

RECOVER versus RESTORE

The RECOVER command does not work with the RESTORE command. You must use the RESTORE command with any files you backed up prior to installing DOS 6.

Note: RESTORE works only with backup files that were created prior to DOS 6.

Related Information

For information about checking your disk for bad sectors, see command "CHKDSK" on page 34.

REM

Enables you to include comments (remarks) in a batch file or in your CONFIG.SYS file.

DOS ignores any batch command or CONFIG.SYS line that begins with REM.

Type

Batch, CONFIG.SYS

Syntax

rem [*comment*]

Parameters

comment Specifies any string of characters you want to include as a comment.

Examples

The following example shows a batch file that uses remarks for both explanations and vertical spacing:

```
@echo off
rem  This batch program formats and checks new disks.
rem  It is named CHECKNEW.BAT.
rem
echo Insert new diskette in drive B.
pause
format b: /v
chkdsk b:
```

Suppose you want to include in your CONFIG.SYS file an explanatory comment before the COUNTRY command. To do this, add the following lines to CONFIG.SYS:

```
rem Set country code to France
country=033
```

Related Information

For information about displaying messages, see command “ECHO” on page 94.

RENAME (REN)

Changes the name of a file or files.

You can rename all files matching the specified file name. You cannot use the RENAME command to rename files across drives, to move files to a different directory location, or to rename subdirectories.

Type

DOS, Internal, Network

Syntax

rename [*drive*:][*path*]*filename1 filename2*

Parameters

[*drive*:][*path*]*filename1*

Specifies the location and name of the file or set of files you want to rename.

filename2

Specifies the new name for the file or, if you use wildcards, the new names for the files. (You cannot specify a new drive or path.)

Notes

Using wildcards with rename

You can use wildcards (* and ?) in either file name parameter. If you use wildcards in *filename2*, the characters represented by the wildcards will be identical to the corresponding characters in *filename1*.

Examples

Suppose you want to change the extensions of all the file names in the current directory that have the extension .TXT to .DOC extensions. To make this change, type the following command:

```
ren *.txt *.doc
```

To rename a file named CHAP10 (on drive B) to PART10, type the following command:

```
ren b:chap10 part10
```

The newly renamed file PART10 remains on drive B.

Related Information

For information about renaming a disk, see command "LABEL" on page 143.

For information about copying files to a different drive or directory, see command "COPY" on page 47.

For information about copying entire directories to a new location, see command "XCOPY" on page 240.

REPLACE

Replaces files in the destination directory with files in the source directory that have the same name. You can also use the REPLACE command to add unique file names to the destination directory.

Type

DOS, External, Network

Syntax

replace [*drive1*:][*path1*]*filename* [*drive2*:][*path2*][**/p**] [**/r**] [**/s**][**/w**] [**/a**|**/u**]

Parameters

[*drive1*:][*path1*]*filename*
Specifies the location and name of the source file or set of files.

[*drive2*:][*path2*]
Specifies the location of the destination file. You cannot specify a file name for files you replace. If you specify neither a drive nor a directory, the Replace program uses the current drive and directory as the destination.

Switches

/a Adds new files to the destination directory instead of replacing existing files. You cannot use this switch with the **/s** or **/u** switch.

/p Prompts you for confirmation before replacing a destination file or adding a source file.

/r Replaces read-only files as well as unprotected files. If you do not specify this switch but attempt to replace a read-only file, an error results and stops the replacement operation.

/s Searches all subdirectories of the destination directory and replaces matching files. You cannot use the **/s** switch with the **/a** switch. The REPLACE command does not search subdirectories specified in *path1*.

/w Waits for you to insert a diskette before the Replace program begins to search for source files. If you do not specify **/w**, The Replace program begins replacing or adding files immediately after you press **Enter**.

/u Replaces (updates) only those files on the destination directory that are older than those in the source directory. You cannot use the **/u** switch with the **/a** switch.

Notes

Replacing files on a diskette drive system

If you have a diskette drive system and need to switch diskettes during the REPLACE operation, you can specify the **/w** switch so that the Replace program will wait for you to switch disks, as necessary.

You cannot use the REPLACE command to update hidden files or system files such as IBMBIO.COM and IBMDOS.COM. For information about changing hidden and system attributes, see command "ATTRIB" on page 26.

You can use the *errorlevel* parameter on the IF command line in a batch program to process exit codes returned by the Replace program.

Examples

Suppose that several directories on drive C contain different versions of a file named PHONES.CLI, which contains client names and phone numbers. To replace all of these files with the latest version of the PHONES.CLI file from the disk in drive A, type the following command:

```
replace a:\phones.cli c:\ /s
```

Suppose you want to add new printer device drivers to a directory on drive C named TOOLS, which already contains several printer device-driver files for a word processor. To do this, type the following command:

```
replace a:*.prd c:\tools /a
```

This command searches the current directory on drive A for any files that have the extension .PRD and then adds these files to the TOOLS directory on drive C. Because the **/a** switch is included, the Replace program adds only those files from drive A that do not exist on drive C.

Replace exit codes

The following list shows each exit code and a brief description of its meaning:

0	REPLACE successfully replaced or added the files.
2	REPLACE could not find the source files.
3	REPLACE could not find the source or destination path.
5	The user does not have access to the files specified to be replaced.
8	There is insufficient system memory to carry out the command.
11	The wrong command line syntax was used.

Related Information

For information about changing file attributes, see command “ATTRIB” on page 26.

RESTORE

Restores files that were backed up by using the BACKUP command provided in DOS prior to this version of DOS.

Note: Restore **does not** work with the CPBACKUP command that is provided with the latest version. It works only with backed up files created with the BACKUP command in prior DOS versions.

You can restore files from similar or dissimilar disk types.

Type

DOS, External, Network

Syntax

restore *drive1*: *drive2*:[*path*[*filename*]][**/s**] [**/p**] [**/b:***date*][**/a:***date*][**/e:***time*][**/l:***time*] [**/m**] [**/n**] [**/d**]

Parameters

drive1: Specifies the drive on which the backed-up files are stored.

drive2: Specifies the drive to which the backed-up files will be restored.

path Specifies the directory to which the backed-up files will be restored. You must specify the same directory from which the files were backed up.

filename
Specifies the names of the backed-up files you want to restore.

Switches

/s Restores all subdirectories.

/p Prompts you for permission to restore files that are read-only (that have the read-only attribute set) or that have changed since the last backup (that have the archive attribute set).

/b:*date*
Restores only those files last modified on or before the specified date. The format of *date* varies according to the COUNTRY setting in your CONFIG.SYS file. For information about specifying *date*, see command "DATE" on page 61.

/a:*date*
Restores only those files last modified on or after the specified date. The format of *date* varies according to the COUNTRY setting in your CONFIG.SYS file. For information about specifying *date*, see command "DATE" on page 61.

/e:*time*
Restores only those files last modified at or earlier than the specified time. The format of *time* varies according to the COUNTRY setting in your CONFIG.SYS file. For information about specifying *time*, see command "TIME" on page 228.

/l:*time* Restores only those files last modified at or later than the specified time. The format of *time* varies according to the COUNTRY setting in your CONFIG.SYS file. For information about specifying *time*, see command "TIME" on page 228.

/m Restores only those files modified since the last backup.

Note: Restore ***does not*** work with the CPBACKUP command that is provided with DOS. It works only with backed up files created with the BACKUP command in DOS versions prior to this version of DOS.

/n Restores only those files that no longer exist on the destination disk.

/d Displays a list of the files on the backup disk that match the names specified in *filename* without restoring any files. Even though no files are being restored, you must specify *drive2* when you use **/d**.

Notes

Checking restored files

Once a file has been restored, you can use the DIR or TYPE command to make sure the file was restored properly.

Limitations on restore

You cannot use the RESTORE command to restore system files (IBMBIO.COM and IBMDOS.COM). The Restore program does not work with drives that have been redirected with the ASSIGN, JOIN, or SUBST command.

Compatibility with previous versions of BACKUP

The DOS RESTORE command restores only files that were backed up by using the DOS BACKUP command available only in previous versions of DOS.

Restore exit codes

The following list shows each exit code and a brief description of its meaning:

0	The Restore program successfully restored the file or files.
1	The Restore program could not find the files to restore.
3	The user pressed **Ctrl+C** to stop the restoring operation.
4	The Restore program stopped because of an error.

You can use the *errorlevel* parameter on the IF command line in a batch program to process exit codes returned by RESTORE.

RMDIR (RD)

Deletes (removes) a directory.

Before you can delete a directory, you must delete its files and subdirectories; the directory must be empty except for the "**.**" and "**..**" symbols.

Type

DOS, Internal, Network

Syntax

rmdir [*drive*:]*path*

rd [*drive*:]*path*

Parameters

[*drive*:]*path*
Specifies the location and name of the directory you want to delete.

Notes

Cannot delete directory with hidden or system files

You cannot delete a directory that contains files, including hidden or system files. If you attempt to do so, DOS displays the following message:

```
Invalid path, not directory,
or directory not empty
```

Use the DIR command to list hidden and system files and the ATTRIB command to remove hidden and system attributes from files. For more information about these commands, see commands "DIR" on page 72 and "ATTRIB" on page 26.

Using the backslash character with the *path* parameter

If you insert a backslash (\) before the first directory name in *path*, DOS treats the directory as a subdirectory of the root directory—regardless of your current directory. If you do not insert a backslash before the first directory name in *path*, DOS treats the directory as a subdirectory of the current directory.

Deleting the current directory

You cannot use RMDIR to delete the current directory. You must first change to a different directory (not a subdirectory of the current directory) and then use RMDIR with a path. If you attempt to delete the current directory, DOS displays a message in the following format:

```
Attempt to remove current directory - drive:path
```

DOS also displays this message if you attempt to delete a directory that has been redirected by using the SUBST command.

Examples

To delete a directory named \USER\SMITH, first ensure that the directory is empty, as in the following example:

```
dir \user\smith /a
```

DOS should display only the "." and ".." symbols.

Then, from any directory except \USER\SMITH, type the following command:

```
rmdir \user\smith
```

Related Information

For information about creating a directory, see command "MKDIR (MD)" on page 157 and command "SUBST" on page 224.

For information about hidden files, see command "ATTRIB" on page 26.

For additional information about removing directories, see "DELTREE" on page 68.

SCHEDULE

Starts the Schedule program. The Schedule program is a full screen utility that enables you to specify a future date and time to automatically run DOS programs. Scheduling is especially useful for lengthy procedures that do not require your presence.

Note: The CPSCHED terminate-and-stay resident program must be loaded for scheduled events to run at the preset date and time.

Type

DOS, External

Syntax

SCHEDULE [*filename*][/25][/28][/43][/50][/60][/in][/bw][/mono][/lcd][/ff][/bf][/nf][/bt][/ngm][/le][/im][/ps2]

Parameters

filename Displays events for the specified file.

Switches

/25 Set screen display to 25 lines (default).

/28 Set screen display to 28 lines (VGA only).

/43 Set screen display to 43 lines (VGA and EGA).

/50 Set screen display to 50 lines (VGA only).

/60 Set screen display to 60 lines (Video 7 only).

/in Run program in color even if color display device is not detected.

/bw Use black and white color scheme.

/mono Use monochrome color scheme (IBM monochrome).

/lcd Use LCD color scheme (for laptops).

/ff Speed up display - can cause “snow” on some display devices (CGA only).

/bf BIOS font (use if graphics does not display properly).

/nf No fonts - do not use graphics characters.

/bt Allow graphics mouse in Windows, allow graphics fonts with Desqview or UltraVision.

/ngm No graphics mouse pointer - use text character mouse pointer.

/le Left handed mouse - exchange left and right mouse buttons.

/im Disable mouse.

/ps2 Reset mouse hardware. (Use if your mouse disappears from the screen or cannot be moved on the screen.)

Notes

The scheduler can invoke programs daily, one time only, workdays only, weekly, monthly-fixed day, monthly-fixed weekday, and bi-weekly. The Schedule program also enables you to change, add, or delete events that have been previously scheduled.

The memory-resident program, CPSCHED, must be active in order to launch an event at the scheduled time. You are notified if CPSCHED is not loaded.

Examples

To start the scheduler type:

```
schedule
```

SET

Displays, sets, or removes DOS environment variables.

You use environment variables to control the behavior of some batch files and programs and to control the way DOS appears and works. The SET command is often used in the AUTOEXEC.BAT file to set environment variables each time you start DOS.

Type

DOS, Internal, Network

Syntax

set [*variable*=[*string*]]

To display the current environment settings, use the following syntax:

set

Parameters

variable
Specifies the variable you want to set or modify.

string Specifies the string you want to associate with the specified variable.

Notes

Displaying the current environment settings

When you type the SET command alone, DOS displays the current environment settings. These settings usually include the *COMSPEC* and *PATH* environment variables that DOS uses to help find programs on disk. *Prompt* and *Dircmd* are two other environment variables that DOS uses. For more information about *Dircmd*, see command “DIR” on page 72.

Using parameters

When you use a SET command and specify values for both *variable* and *string*, DOS adds the specified *variable* value to the environment and associates *string* with that variable. If *variable* already exists in the environment, the new *string* value replaces the old *string* value.

If you specify only a variable and an equal sign (without a string) for the SET command, DOS clears the *string* value associated with the variable (as if the variable is not there at all).

Using set in batch files

When creating batch files, you can use the SET command to create variables and use them in the same way as you would the numbered variables *%0* through *%9*. You can also use the variables *%0* through *%9* as input for the SET command.

Calling a SET variable from a batch file

When you call a *variable* value from a batch file, you must enclose the value with percent signs (%). For example, if your batch program creates an environment variable named *Baud*, you can use the string associated with *Baud* as a replaceable parameter by inserting *%baud%* on the command line.

Effect of SET on environment space

After you use a SET command, DOS might display the following message:

```
Out of environment space
```

This message means the available environment space is insufficient to hold the new variable definition. For information about how to increase the environment space, see command "COMMAND" on page 40.

Examples

To set an environment variable named *Include* so that the string C:\INC (the INC directory on drive C) is associated with it, type the following command:

```
set include=c:\inc
```

You can then use the string C:\INC in batch files by enclosing the name *include* with percent signs (%). For example, you might include the following command in a batch file in order to display the contents of the directory associated with the INCLUDE environment variable:

```
dir %include%
```

When DOS processes this command, the string C:\INC replaces *%include%*.

Another possible use for the SET command is in a batch program that adds a new directory to the *PATH* environment variable, as the following example shows:

```
@echo off
rem ADDPATH.BAT adds a new directory
rem to the PATH environment variable.
set path=%1;%path%
set
```

Related Information

For information about setting environment variables that DOS uses to control its own operations, see commands "PATH" on page 175, "PROMPT" on page 191, and "DIR" on page 72.

SETUP

This command is used only when you are installing the latest version of DOS. It is used when the Install diskette #1 is in drive A. You can receive the online help by typing `setup /?` at the command prompt.

Type

DOS

Syntax

setup [**/a**] [**/b**] :[**/e**] [**/p**][**/t:***filepath*][**/u**] [**/w**]

Switches

/a Specifies the LAN Server Administrator installation option.

/b Specifies to use black and white instead of color screen display.

/e Use to install DOS tools without copying DOS files.

/p Use to install on a disk that might be incompatible with DOS.

/t:*filepath* Specifies the target path to copy DOS files to.

/u Use to uninstall the latest version of DOS.

/w Allows Windows tools to be installed even though a valid Windows 3.1 directory was not found.

Related Information

For more information about installing this version of DOS, see the *PC DOS Installation Guide*.

SETVER

Sets the DOS version number that is reported to a program.

You can also use this command to display and modify the *version table*, that lists names of programs and the number of the DOS version with which they are designed to run. If you are using a program that has not been updated for this version of DOS , you can add its name to the version table by using the SETVER command.

Type

DOS, External, Network

Syntax

To display the current version table, type:

setver [*drive:path*]

To add an entry, type

setver [*drive:path*][*filename n.nn*]

To delete an entry, type:

setver [*drive:path*][*filename* **[/delete [/quiet]]]**

Parameters

[*drive*:*path*]
Specifies the location of the SETVER.EXE file.

filename
Specifies the name of the program file that you want to add to the version table. You cannot use a wildcard (* or ?).

n.nn Specifies the DOS version (for example, 3.3 or 4.01) that DOS reports to the specified program file.

Switches

/delete Deletes the version-table entry for the specified program file. You can abbreviate this switch as **/d**.

/quiet Hides the message typically displayed during deletion of an entry from the version table.

Notes

Loading the version table into memory

Before you can use the SETVER command, the version table must be loaded into memory by a DEVICE command in your CONFIG.SYS file. By default, the DOS Setup program modifies your CONFIG.SYS file to ensure that the version table is loaded into memory each time you start your system.

Setting the version number of a command interpreter

You can use SETVER to set the version number of a command interpreter; however, if you set the version number of the command interpreter (COMMAND.COM) for this version of DOS, you might not be able to start your system.

Using the version table

Many programs designed to run with a previous version of DOS will run correctly with the latest version of DOS. In some cases, however, a program might not run correctly unless its name is included in the version table. The table indicates to the program that it is running with the DOS version for which it was designed, even though it is running with the latest version of DOS. By interpreting the latest version of DOS as the earlier version, the program will probably run correctly; however, using the SETVER command will not solve the problem if the program is not compatible with the latest version of DOS.

SETVER confirmation If you make changes to the version table and no errors are detected, DOS displays the following message:

```
WARNING - The application you are adding to the IBM DOS version table
may not have been verified by IBM on this version of IBM DOS.
Please contact your software vendor for information on whether this
application will operate properly under this version of IBM DOS.
If you execute this application by instructing IBM DOS to report a
different IBM DOS version number, you might lose or corrupt data, or
cause system instabilities.  In that circumstance, IBM is not
responsible for any loss or damage.

Version table successfully updated
The version change will take effect the next time you restart your system
```

Restarting after updating the version table

When you update the version table by adding or deleting entries, you must restart your system before the changes will take effect.

Updating existing entries

If you specify a file name that is already in the version table, the new entry replaces the existing entry.

Setver exit codes

The following list shows each exit code and a brief description of its meaning:

0 SETVER successfully completed its task.

1 The user specified an invalid command switch.

2 The user specified an invalid file name.

3 There is insufficient system memory to carry out the command.

4 The user specified an invalid version-number format.

5 SETVER could not find the specified entry in the version table.

6 SETVER could not find the SETVER.EXE file.

7 The user specified an invalid drive.

8 The user specified too many command-line parameters.

9 SETVER detected missing command-line parameters.

10 SETVER detected an error while reading the SETVER.EXE file.

11 The SETVER.EXE file is corrupt.

12 The specified SETVER.EXE file does not support a version table.

13 There is insufficient space in the version table for a new entry.

14 SETVER detected an error while writing to the SETVER.EXE file.

You can use the **errorlevel** parameter on the IF command line in a batch program to process exit codes returned by the SETVER process.

Examples

Suppose you have a program file named MYPROG.EXE that runs with DOS version 3.30. To run MYPROG.EXE, you must first use the SETVER command to create an entry in the version table that will cause MYPROG.EXE to interpret the latest version of DOS as version 3.30:

```
setver myprog.exe 3.30
```

To delete the MYPROG.EXE entry from the version table (without otherwise affecting the MYPROG.EXE file), type the following command:

```
setver myprog.exe /delete
```

To list the contents of the current version table, type the following command:

```
setver
```

DOS displays two columns: the left column lists the names of the program files; the right column lists the corresponding DOS version with which each file is set to run.

SETVER.EXE

This is one of the installable device drivers provided with DOS. It loads the DOS version table into memory.

Related Information

For a detailed description about this device driver, see "SETVER.EXE" on page 282.

SHARE

Starts the Share program, which installs file-sharing and locking capabilities on your hard disk.

Type

DOS, External, Network

Syntax

share [/**f:***space*] [/**l:***locks*]

In your CONFIG.SYS file, use the following syntax:

install=[[*drive*:]*path*]*share.exe* [/**f:***space*] [/**l:***locks*]

Parameters

[*drive*:]*path*

Specifies the location of the SHARE.EXE file.

Switches

/**f:***space*

Allocates file space (in bytes) for the DOS storage area used to record file-sharing information. The default value is 2048.

/**l:***locks*

Sets the number of files that can be locked at one time. The default value is 20.

Notes

Common use of SHARE

Typically, you use SHARE in a network or multi-tasking environment in which programs share files. SHARE loads the code that supports file-sharing and locking in these environments. Once you install SHARE, DOS uses the code loaded by SHARE to validate all read and write requests from programs.

Allocating space for file-sharing information

When deciding how many bytes to allocate for file sharing, note that each open file requires enough space for the length of the full path and file name. The average length of a file name and its path is 20 bytes.

Examples

The following example shows how you can use the INSTALL command in your CONFIG.SYS file to load SHARE, with the default values for the **/f** and **/l** switches. DOS searches for the file SHARE.EXE in the DOS directory on drive C.

```
install=c:\dos\share.exe
```

The following example allocates 4096 bytes for storing file-sharing information and specifies that 25 files can be locked at one time. Again, DOS searches for SHARE.EXE in the DOS directory on drive C:

```
install=c:\dos\share.exe /f:4096 /l:25
```

SHELL

Specifies the name and location of the command interpreter you want DOS to use.

If you want to use your own command interpreter (instead of COMMAND.COM), you can specify its name by adding a **shell** command to your CONFIG.SYS file.

Type

CONFIG.SYS

Syntax

shell=[[*drive*:]*path*]*filename* [*parameters*]

Parameters

[[*drive*:]*path*]*filename*
Specifies the location and name of the command interpreter you want DOS to use.

parameters
Specifies any command-line parameters or switches that can be used with the specified command interpreter.

Notes

Default setting

The default command interpreter for DOS is COMMAND.COM. If you do not use a SHELL command in your CONFIG.SYS file, DOS searches for COMMAND.COM in the root directory of your startup drive. You need to use the SHELL command if you want to specify a COMMAND.COM file that is not in the root directory or if you do not want to use the default environment size for COMMAND.COM. For information about COMMAND.COM switches, see command “COMMAND” on page 40.

Using switches with a command interpreter

The SHELL command itself does not accept any switches, but if the command interpreter does, you can include them on the SHELL command line.

Examples

Suppose the file NEWSHELL.COM is in a directory named BIN on your startup drive, and suppose you want to use NEWSHELL.COM as your command interpreter. To do this, add the following command to your CONFIG.SYS file:

```
shell=\bin\newshell.com
```

Suppose you add the line **shell=newcmdp.com** to your CONFIG.SYS file, and suppose the NEWCMDP.COM command interpreter accepts the switches **/c**, **/p**, and **/e**. You can now use any of these switches on the **shell** command line. Thus, the following command would be valid:

```
shell=newcmdp.com /c /p /e
```

The SHELL command is the preferred method of using the Command command to increase the size of the environment. To increase the environment size to 512 bytes, add the following command to your CONFIG.SYS file:

```
shell=command.com /e:512 /p
```

To start a DOS command interpreter located in the directory OLD on drive C, add the following command to your CONFIG.SYS file:

```
shell=c:\old\command.com c:\old /e:256 /p
```

SHIFT

Changes the position of replaceable parameters in a batch file.

Type

Batch, Internal

Syntax

shift

Notes

How the SHIFT command works

The SHIFT command changes the values of the replaceable parameters *%0* through *%9*, by copying each parameter into the previous one. In other words, the value of *%1* is copied to *%0*, the value of *%2* is copied to *%1*, and so on. This is useful for writing a batch file that performs the same operation on any number of parameters.

Working with more than 10 command-line parameters

You can also use the SHIFT command to create a batch file that can accept more than 10 parameters. If you specify more than 10 parameters on the command line, those that appear after the tenth (*%9*) will be shifted one at a time into *%9*.

Shifting parameters back

There is no backward SHIFT command. Once you carry out the SHIFT command, you cannot recover the first parameter (*%0*) that existed before the shift.

Examples

The following batch file, MYCOPY.BAT, shows how to use the SHIFT command with any number of parameters. It copies a list of files to a specific directory. The parameters are the directory name followed by any number of file names.

```
@echo off
rem MYCOPY.BAT copies any number of files
rem to a directory.
rem The command uses the following syntax:
rem mycopy dir file1 file2 ...
set todir=%1
:getfile
shift
if “%1"==”" goto end
copy %1 %todir%
goto getfile
:end
set todir=
echo All done
```

SMARTDRV

Creates a disk cache in extended or expanded memory.

A disk cache can significantly speed up DOS disk operations. You can control the size of the SMARTDRV memory cache, and you can set up the disk cache in extended memory.

Normally, it is started by the SMARTDRV command in your AUTOEXEC.BAT file. If you have a SCSI (small computer system interface) hard disk, you might need to use the double-buffering feature of SMARTDRV, which requires a DEVICE command in your CONFIG.SYS file as well.

For additional information about the SMARTDRV.EXE device driver, see "SMARTDRV.EXE" on page 282.

Note: SMARTDRV is a terminate-and-stay resident program and a device driver. The syntax and parameters are different than they were in previous DOS versions.

Type

DOS, External

Syntax

When starting SMARTDRV from your AUTOEXEC.BAT file or from the command prompt, use the following syntax:

smartdrv [[*drive*[+|-]]...][/**e:***ElementSize*] [*InitCacheSize*][*WinCacheSize*]][/**b:***BufferSize*][/**c**] [/**r**][/**l**][/**q**][/**s**]

After SMARTDRV is running, use the following syntax:

smartdrv [[*drive*[+|-]]...][/**c**][/**r**]

Parameters

[*drive*]

Specifies the letter of the disk drive for which you want to control caching. If you do not specify a drive letter, diskette drives will be read-cached but not write-cached, hard disk drives will be both read-cached and write-cached, and CD-ROM, compressed drives, and network drives will be ignored.

You can specify multiple disk drives.

[+|-] Enables (+) or disables (-) caching. Use the plus (+) and minus (-) signs to override the default settings. If you specify a drive letter without a plus or minus sign, read-caching is enabled and write-caching is disabled. If you specify a drive letter followed by a plus sign (+), read-caching and write-caching are enabled. If you specify a drive letter followed by a minus sign (-), both read-caching and write-caching are disabled.

InitCacheSize

Specifies the cache size in kilobytes when SMARTDRV starts (before Windows is running). The size of the disk cache affects how efficiently SMARTDRV runs. In general, the larger the cache, the less often SMARTDRV needs to read information from the disk, which speeds up the performance of your system. If you do not specify an InitCacheSize value, SMARTDRV sets the value according to how much memory your system has (see Table 1 on page 219).

WinCacheSize

Limits how much Windows can reduce the cache size. Windows reduces the size of the cache to recover memory for its own use. Windows and SMARTDRV then cooperate to provide optimum use of your system memory. When you quit Windows, the cache is restored to its normal size. WinCacheSize specifies the smallest size to which Windows can reduce the cache. The default value depends on how much available memory your system has (see Table 1 on page 219). If you specify a value for InitCacheSize that is smaller than the value specified for WinCacheSize, InitCacheSize is set to the same size as WinCacheSize.

Switches

/**b**:*BufferSize* Specifies the size of the read-ahead buffer. A read-ahead buffer is additional information that SMARTDRV reads when an application reads information from the hard disk. For example, if an application reads 512K of information from a file, SMARTDRV then reads the amount of information specified by *BufferSize* and saves it in memory. The next time the application needs to read information from that file, it can read it from memory instead. The default size of the read-ahead buffer is 16K. Its value can be any multiple of *ElementSize*.

/**c** Writes all cached information from memory to the hard disk. SMARTDRV writes information from memory to the hard disk when other disk activity has slowed. You might use this option if you are going to turn off your computer and want to make sure all cached information has been written to the hard disk.

/e:*ElementSize* Specifies in bytes the amount of the cache that SMARTDRV moves at a time. Valid values are 1024, 2048, 4096, and 8192. The default value is 8192.

/r Clears the contents of the existing cache and restarts SMARTDRV.

/l Loads SMARTDRV into conventional memory and prevents SMARTDRV from loading into upper memory blocks (UMBs), even if there are UMBs available. You can use this option if upper memory is enabled for use by programs.

/q Prevents SMARTDRV from displaying error and status messages when it starts.

/s Displays additional information about the status of SMARTDRV.

The following table shows what the default values for InitCacheSize and WinCacheSize will be, depending on the amount of available extended memory your computer has.

Table 1. InitCacheSize and WinCacheSize cache default values

Extended Memory	InitCacheSize	WinCacheSize
Up to 1 MB	All extended memory (minus 64K)	Zero (no caching)
Up to 2 MB	1MB	256K
Up to 4MB	1MB	512K
Up to 6 MB	2 MB	1 MB
6 MB or more	2 MB	2 MB

CAUTION:
Ensure that SMARTDRV has completed all writing-cache before you turn off your computer. (This is not necessary if you restart your computer by pressing Ctrl+Alt+Del.) To make sure this has happened, type `smartdrv /c` at the command prompt. After all disk activity has stopped, you can safely turn off your computer.

Notes

Using extended memory

For SMARTDRV.EXE to use extended memory, you must first install HIMEM.SYS or another extended-memory manager that conforms to the Lotus/Intel/Microsoft/AST eXtended Memory Specification (XMS).

Loading SMARTDRV into the upper memory area

If the Upper Memory Blocks are available, the SMARTDRV program automatically loads into upper memory. You do not have to use LOADHIGH with SMARTDRV.EXE.

Using SMARTDRV with Windows

The SMARTDRV command should not be used after Windows has started.

Examples

To create a disk cache in extended memory and set a cache size of 256K (the default size), add the following line to your AUTOEXEC.BAT file:

```
c:\dos\smartdrv.exe
```

Suppose you want to create a disk cache in extended memory, allocate a cache size of 2048K, and ensure that programs cannot reduce the size of the cache to less than 1024K. To do this and to specify that SMARTDRV.EXE is located in the DOS directory on drive C, add the following line to your AUTOEXEC.BAT file:

```
c:\dos\smartdrv.exe 2048 1024
```

SMARTDRV.EXE

This is one of the installable drivers provided by DOS. It is used only if you have a need for double-buffering. See “SMARTDRV.EXE” on page 282 for more information.

SORT

Reads input, sorts data, and writes the results to the screen, to a file, or to another device.

SORT acts as a filter, reading characters in a specified column and rearranging them in ascending or descending order.

Type

DOS, External, Network

Syntax

sort [/**r**] [/+*n*][<] [*drive1*:] [*path1*]*filename1* [> [*drive2*:][*path2*]*filename2*] [*command* |]
sort[/**r**][/+*n*][> [*drive2*:][*path2*]*filename2*]

Parameters

[*drive1*:][*path1*]*filename1*
Specifies the location and name of the file whose data you want to sort.

[*drive2*:][*path2*]*filename2*
Specifies the location and name of a file in which the sorted output is to be stored.

command
Specifies a command whose output is the data you want to sort.

Switches

/r Reverses the order of the sorting operation; that is, sorts from Z to A, and then from 9 to 0.

/+*n* Sorts the file according to the character in column *n*. If you do not use this switch, the SORT command sorts data according to the characters in column 1.

Notes

Specifying a source

Unless you specify the *command* or *filename* parameter, SORT acts as a filter and takes input from the DOS standard input (usually from the keyboard, from a pipe (|), or from a file).

Using redirection symbols with sort

You can use the pipe or the less-than sign (<) to direct data through the SORT command from *command* or *filename*. If you want to display the information one screen at a time or direct the information to a file, you can also specify the MORE command or a file name. You can use the greater-than sign (>) to direct the sorted output to a file.

Collating sequence

SORT uses the collating-sequence table corresponding to the country code and code-page settings. Characters greater than ASCII code 127 are sorted based on information in the COUNTRY.SYS file or in an alternate file specified by the COUNTRY command in your CONFIG.SYS file.

Uppercase vs. lowercase

SORT does not distinguish between uppercase and lowercase letters.

Limits on file size

The SORT command can handle files as large as 64K.

Examples

The following command reads the file EXPENSES.TXT, sorts it in reverse order, and displays it on your screen:

```
sort /r < expenses.txt
```

Suppose you want to search a large file named MAILLST.TXT for the text “Jones”, and suppose you want to sort the results of the search. To do this, use the pipe (|) to direct the output of a FIND command to the SORT command, as shown in the following example:

```
find “Jones” maillst.txt | sort
```

The command produces a sorted list of lines that contain the specified text.

To sort keyboard input and display the results alphabetically on the screen, you can first use the SORT command with no parameters, as the following example shows:

```
sort
```

Then type the text you want sorted, pressing **Enter** at the end of each line. When you have finished typing text, press **Ctrl+Z**, and then press **Enter**. The SORT command displays the text you typed, sorted alphabetically. You could also redirect sorted keyboard input to a file.

Related Information

For information about displaying information one screen at a time, see command "MORE" on page 168.

STACKS

Supports the dynamic use of data stacks to handle hardware interrupts.

Type

CONFIG.SYS

Syntax

stacks=*n*,*s*

Parameters

n Specifies the number of stacks. Valid values for *n* are 0 and numbers in the range 8 through 64.

s Specifies the size (in bytes) of each stack. Valid values for *s* are 0 and numbers in the range 32 through 512.

Notes

Default settings

The default settings for the **stack** command are as follows:

Computer	**Stacks**
IBM PC, IBM PC/XT, IBM PC-Portable	0,0
Other	9,128

Special cases for stack allocation

Upon receiving a hardware interrupt, DOS allocates one stack from the specified number of stacks. When you specify 0 for the *n* and *s* values, DOS allocates no stacks. If the values are 0, each running program must have enough stack space to accommodate the computer's hardware interrupt drivers. Many computers operate correctly, saving some memory for programs, with *n* and *s* values of 0. If, however, your computer becomes unstable when you set these values to 0, return to the default values.

Examples

To allocate 8 stacks of 512 bytes each for hardware-interrupt handling, add the following command to your CONFIG.SYS file:

```
stacks=8,512
```

SUBMENU

Defines an item on a startup menu that, when selected, displays another set of choices. You can use this command only within a menu block in your CONFIG.SYS file.

Type

CONFIG.SYS

Syntax

submenu=*blockname* [*,menu_text*]

Parameters

blockname Specifies the name of the associated menu block. The menu block must be defined elsewhere in the CONFIG.SYS file.

menu_text Specifies the text you want DOS to display for the menu item. If you do not display any menu text, DOS displays the block name as the menu item.

Notes

Submenu names

Unlike the main menu block, which must have the block name [MENU], a menu block for the submenu can have any name you want. If DOS cannot find a block with the name you have specified, the item will not appear on the startup menu.

Length of Submenu blocknames

The block name can be up to 70 characters long and can contain most printable characters.

Length of *menu_text* parameter

The menu text can be up to 70 characters long and can contain any characters you want.

Related Information

For more information about menu blocks and multiple configurations, see the *PC DOS User's Guide*.

For information about the [MENU] command, see command "[MENU]" on page 151.

SUBST

Associates a path with a drive letter.

The drive letter you assign represents a virtual drive, because you can use the drive letter in commands as if it represented a physical drive.

Type

DOS, External

Syntax

subst [*drive1*: [*drive2*:]*path*]

subst *drive1*: **/d**

Parameters

drive1:	Specifies the virtual drive to which you want to assign a path.
drive2:	Specifies the physical drive that contains the specified path (if different from the current drive).
path	Specifies the path that you want to assign to a virtual drive.

Switches

/d	Deletes a virtual drive.

Notes

Using other commands with SUBST

The following commands do not work, or should not be used, on drives used in the SUBST command:

ASSIGN
CHKDSK
DISKCOMP
DISKCOPY
FDISK
FORMAT
LABEL
RECOVER
RESTORE
SYS

Valid *drive1* values

The *drive1* parameter must be within the range specified by the LASTDRIVE command. If not, SUBST displays the following error message:

```
   Invalid parameter - drive1:
```

Ensuring compatibility with future versions of DOS

To ensure compatibility with future versions of DOS, you should use SUBST rather than the ASSIGN command.

Examples

To display the names of the virtual drives in effect, use the following syntax:

```
   subst
```

The following command creates a virtual drive Z for the path B:\USER\BETTY\FORMS:

```
   subst z: b:\user\betty\forms
```

Now, instead of typing the full path, you can reach this directory by typing the letter of the virtual drive, followed by a colon, as in the following example:

```
   z:
```

This example works only if you have defined Z as the highest letter that DOS recognizes as a disk drive. The Z is defined by including the line **lastdrive=z** in your CONFIG.SYS file.

Related Information

For information about joining a disk drive to a directory, see command “JOIN” on page 136.

For information about increasing the number of available drive letters, see command “LASTDRIVE” on page 145.

SWITCHES

Provides special options. Use this command only in your CONFIG.SYS file.

Type

CONFIG.SYS

Syntax

switches=[/w][/k] [/n][/f]

Switches

/w Specifies that the WINA20.386 file has been moved to a directory other than the root directory. You need to use this switch only if you are using Windows 3.0 in enhanced mode and have moved the WINA20.386 file from the root directory to another directory.

/k Forces an enhanced keyboard to behave like a conventional keyboard.

/n Prevents you from using the **F5** or **F8** key to bypass startup commands.

/f Skips the delay after displaying the "Starting IBM DOS..." message during startup.

Notes

When to use the switches command

If you have a program that does not correctly interpret input from an enhanced keyboard, add this command to your CONFIG.SYS file so your enhanced keyboard will use conventional keyboard functions.

Using the /k switch with ANSI.SYS

If you use the **switches=/k** command and you install the ANSI.SYS device driver, use the **/k** switch on the DEVICE command line for ANSI.SYS.

Examples

If you want DOS to use conventional keyboard functions even though you are using an enhanced keyboard, add the following command to your CONFIG.SYS file:

```
switches=/k
```

SYS

Copies DOS system files and the DOS command interpreter (COMMAND.COM) to the disk in a drive you specify.

Note: These three files are all retrieved from the same location. The two system files (IBMBIO.COM and IBMDOS.COM) are hidden files and do not typically appear when you type the DIR command.

Type

DOS, External

Syntax

sys [*drive1*:][*path*] *drive2*:

Parameters

[*drive1*:][*path*]
Specifies the location of the system files. If you do not specify a path, DOS searches the root directory on the current drive for the system files.

drive2: Specifies the drive to which you want to copy the system files. These files can be copied only to a root directory, not to a subdirectory.

Notes

How the SYS command copies files

The SYS command copies the files in the following order: IBMBIO.COM, IBMDOS.COM, and COMMAND.COM. The two system files (IBMBIO.COM and IBMDOS.COM) are hidden files and do not typically appear when you type the DIR command.

No requirement for contiguous files

DOS no longer requires the two system files to be contiguous. This means that when you want to copy a new version of DOS to a disk containing system files for DOS Version 3.3 or earlier, you need not reformat the disk.

Using the SYS command on assigned drives and networks

The SYS command does not work on drives that have been redirected by using the ASSIGN, JOIN, or SUBST command. You cannot use *drive2*: as a network drive. However, *drive1*: can be a network drive from which to copy the files.

Examples

To copy the DOS system files and command interpreter from the disk in the current drive to a diskette in drive A, type the following command:

```
sys a:
```

To copy the DOS system files and command interpreter from the disk in drive D to a diskette in drive A, type the following the command:

```
sys d:\ a:
```

Related Information

For information about copying files, see commands “COPY” on page 47 and “XCOPY” on page 240.

TIME

Displays the system time or sets the internal clock of your computer.

DOS uses time information to update the directory whenever you create or change a file.

Type

DOS, Internal, Network

Syntax

time [*hours*:[*minutes*[:*seconds*[.*hundredths*]]][*a*|*p*]]

To display the current time or to display a prompt by which you can change the current time, use the following syntax:

time

Parameters

hours Specifies the hour. Valid values are in the range 0 through 23.

minutes
Specifies minutes. Valid values are in the range 0 through 59.

seconds
Specifies seconds. Valid values are in the range 0 through 59.

hundredths
Specifies hundredths of a second. Valid values are in the range 0 through 99.

a|*p* Specifies A.M or P.M. for the 12-hour time format. If you type a valid 12-hour time but do not type *a* or *p*, TIME uses *a* (for A.M.).

Notes

Changing the time format

You can change the TIME format by changing the COUNTRY setting in your CONFIG.SYS file. For more information, see command "COUNTRY" on page 51. Depending on the country code, DOS will display the time in the 12-hour format or the 24-hour format. If you are setting the time in the 12-hour format, be sure to specify *p* for hours after noon.

Ensuring that DOS prompts you for the time

If you want DOS to prompt you for the current time whenever you start your system, you can add the TIME command to your AUTOEXEC.BAT file. DOS will automatically prompt you for the time and date if you do not have an AUTOEXEC.BAT file.

Examples

To set the clock of your computer to 1:36 P.M., use either of the following commands:

```
time 13:36
```

```
time 1:36p
```

Related Information

For information about changing the current date, see command "DATE" on page 61.

TREE

Graphically displays the directory structure of a path or of the disk in a drive.

Type

DOS, External, Network

Syntax

tree [*drive*:][*path*][/**f**][/**a**]

Parameters

drive: Specifies the drive that contains the disk for which you want to display the directory structure.

path Specifies the directory for which you want to display the directory structure.

Switches

/f Displays the names of the files in each directory.

/a Specifies that TREE is to use text characters instead of graphic characters to show the lines linking subdirectories. Use this switch with code pages that do not support graphic characters and to send output to printers that does not properly interpret the graphic characters.

To display, one screen at a time, the files in all the directories on drive C, type the following command:

```
tree c:\ /f | more
```

To print the same list that the previous example displayed, type the following command:

```
tree c:\ /f > prn
```

Related Information

For information about displaying the contents of a directory, see command “DIR” on page 72.

TYPE

Displays the contents of a text file.

Use the TYPE command to view a text file without modifying it.

Type

DOS, Internal, Network

Syntax

type [*drive*:][*path*]*filename*

Parameters

[*drive*:][*path*]*filename*	Specifies the location and name of the file that you want to view.

Notes

Displaying binary files

If you display a binary file or a file created by a program, you might see strange characters on the screen, including form-feed characters and escape-sequence symbols. These characters represent control codes used in the binary file. In general, you should avoid using the TYPE command to display binary files.

Examples

If you want to display the contents of a file named HOLIDAY.MAR, type the following command:

```
type holiday.mar
```

If the file you want to display is long, you can use the MORE command along with TYPE as shown in the following command, to display the contents of the file one screen at a time:

```
type holiday.mar | more
```

Related Information

For information about displaying file names and file sizes, see command “DIR” on page 72.

For information about displaying text files one screen at a time, see command “MORE” on page 168.

UMBCGA.SYS

This is one of the installable device drivers provided with DOS. See "UMBCGA.SYS" on page 284 for more details.

UMBEMS.SYS

This is one of the installable device drivers provided with DOS. See "UMBEMS.SYS" on page 286 for more details.

UMBHERC.SYS

This is one of the installable device drivers provided with DOS. See "UMBHERC.SYS" on page 285 for more details.

UMBMONO.SYS

This is one of the installable device drivers provided with DOS. See "UMBMONO.SYS" on page 283 for more details.

UNDELETE

UNDELETE is a full-screen utility that restores files previously deleted.

Type

DOS, External

Syntax

undelete [[*drive*:][*path*]][*filename*][**/s**|**/dt**|**/dos**|**/all**][**/nc**][**/list**][**/purge**][**/purgeall**][**/video**]

Parameters

[*drive*:][*path*]*filename*

Specifies the location and name of the file or set of files you want to recover. By default, UNDELETE restores all deleted files in the current directory.

Switches

/s Recovers only those files protected by the Delete Sentry method of delete protection.

/dt Recovers only those files protected by the Delete Tracker method of delete protection.

/dos Recovers only those files protected by DOS.

/all Recovers all deleted files.

/nc Recovers files without asking for confirmation for each file.

/list Lists the deleted files that are available to be recovered in the specified directory.

/purge Purges the deleted files in the specified directory.

/purgeall Purges all files on the specified drive.

/video Displays help for the following video and mouse selections:

- **/25** Sets the screen display to 25 lines (the default).
- **/28** Sets the screen display to 28 lines (VGA only).
- **/43** Sets the screen display to 43 lines (VGA and EGA).
- **/50** Sets the screen display to 50 lines (VGA only).
- **/60** Sets the screen display to 60 lines (Video 7 only).
- **/in** Runs the program in color even if a color display is not detected.
- **/bw** Specifies the use of the black and white scheme.
- **/mono** Specifies the use of the IBM monochrome color scheme.
- **/lcd** Specifies the use of the LCD color scheme (for laptops).
- **/ff** Speeds up the display.
- **/bf** Specifies the use of the BIOS font. Use it if your graphics does not display properly.
- **/nf** Specifies that fonts are not to be used (do not use graphic characters).
- **/bt** Enables the use of a graphics mouse in Windows, and graphics fonts with Desqview or UltraVision.
- **/ngm** Specifies that only a text character mouse pointer is to be used.
- **/le** Specifies the mouse as left-handed.
- **/im** Disables the mouse.

/ps2 Resets the mouse hardware. (Use if the mouse pointer disappears or freezes.)

Related Information

For more information about Delete Sentry and Delete Tracker, see command "DATAMON" on page 60.

UNFORMAT

Restores a disk erased by the FORMAT command or restructured by the RECOVER command.

UNFORMAT restores only local hard disk drives and diskette drives; it cannot be used on network drives. The UNFORMAT command can also rebuild a corrupted disk partition table on a hard disk drive.

Type

DOS, External

Syntax

unformat *drive*:[**/l**][**/test**][**/p**]

Parameters

drive: Specifies the drive that contains the disk on which you want to recover files.

Switches

/l Lists every file and subdirectory found by UNFORMAT. If you do not specify this switch, UNFORMAT lists only subdirectories and files that are fragmented. To suspend scrolling of the displayed list, press **Ctrl+S**; to resume scrolling, press any key.

/test Shows how UNFORMAT would recreate the information on the disk, but does not actually unformat the disk.

/p Sends output messages to the printer connected to LPT1.

Notes

Limitations on the UNFORMAT command

If the FORMAT command was used with the **/u** switch, UNFORMAT cannot restore the disk to its previous condition.

Unformatting a disk

The UNFORMAT command can restore your disk by using information in the boot directory and file allocation table on the disk.

If you specify the /l switch, UNFORMAT displays how many subdirectories it has found as the disk is rebuilt. All files in each subdirectory are also displayed.

If UNFORMAT finds a file that appears to be fragmented (that is, stored in separate places on the disk), it cannot recover the file because it cannot locate the remaining portions of the file. In this case, the UNFORMAT command prompts you to confirm whether you want UNFORMAT to truncate the file (that is, recover only the first part of the file that it can locate) or delete the file altogether.

If UNFORMAT does not prompt you for a specific file, that file is most likely intact. In certain circumstances, however, UNFORMAT might not recognize that a file is fragmented, even though it has located only a portion of the file. If this happens to a program file, the program does not run properly. If this happens to a data file, information is lost and the program that created the data file might not be able to read it. In these cases, your only recourse is to restore the files from your original diskettes or backup files.

Examples

To restore a formatted disk in drive A, type the following command:

```
unformat a:
```

To determine whether UNFORMAT can restore a formatted disk in drive A, type the following command:

```
unformat a: /test
```

To restore a formatted disk in drive A, listing all files and subdirectories, type the following command:

```
unformat a: /l
```

Related Information

For information about formatting a disk, see command "FORMAT" on page 115.

VER

Displays the DOS version number.

Type

DOS, Internal, Network

Syntax

ver

Examples

When you enter the VER command, DOS displays the following message:

```
IBM DOS
```

VERIFY

Tells DOS to verify that your files are written correctly to a disk.

Type

DOS, Internal, Network

Syntax

verify [on|off]

Switches

on|off Specifies whether DOS is to verify that write operations are done correctly. The *on* value enables this verification process. The *off* value disables it.

Notes

Displaying the current status of the on|off switch

Use the VERIFY command without a switch to find out whether verification is turned on.

How verify affects performance

Turning VERIFY on slows down all disk write operations.

Verifying files as you copy them

To verify that files are copied correctly, you can also use the */v* switch with the COPY or XCOPY command. For more information about the */v* switch, see command "COPY" on page 47 and "XCOPY" on page 240.

Related Information

For information about checking a disk for bad sectors, see command "CHKDSK" on page 34.

VOL

Displays the disk volume label and serial number, if they exist.

A serial number is displayed for a disk formatted with DOS version 4.0 or later.

Type

DOS, Internal, Network

Syntax

vol [*drive*:]

Parameter

drive: Specifies the drive that contains the disk for which you want to display the volume label and serial number.

Notes

To cause DOS to display the volume label of the disk in the current drive, you can use the VOL command with no parameter.

Related Information

For information about assigning a volume label, see commands "FORMAT" on page 115 and "LABEL" on page 143.

WNBACKUP

Makes a backup copy of data to diskettes, tape, or a network drive.

WNBACKUP is a Windows utility that can backup files using full, incremental, differential, or unattended methods. It can compress, and encrypt data as it is backed up.

Type

Windows

Syntax

From the DOS command-line prompt:

win wnbackup [*drive*:][*setup file name*][**/date=***mmddyy-mmddyy*][**/drive=x**[*xx*]]

[**/exattr=***srh*][**/addr=***base-i-d*] [**/rate=***x*][**/full**|**/copy**|**/fullerase**
|**/inc**|**/sep**|**/dif**][**/ecc**|**/noecc**][**/sf**|**/nonsf**][**/st**] [**/nonst**][**/save**|**/nosave**][**/no**]
[**/dcheck**][**/nodcheck**][**/i**][**/b**][**/c**][**/r**[*setup file name*]]

Note: You can use as many options as you need. However, combinations that affect the same option (such as **/full** and **/sep**) are not allowed to be used at the same time. If two mutually exclusive options are specified, the last one on the command line takes effect and the first is ignored.

From the Windows desktop: Double-click on the Backup Icon in the IBM Tools folder.

Parameters

drive Specifies a drive to back up to. If you specify this parameter and a setup file, the drive that you specify will take effect along with any drives specified in the setup file.

setup file name Loads the specified file. To start WNBACKUP with the DAILY setup file, type **win wnbackup daily** at the DOS command-line prompt.

Switches

/date=*mmddyy-mmddyy*
Backs up or restores all files with the dates within the ranges specified.

/drive=*x*[*xx*] Specifies the drive and media to back up to. The *x* can be a drive letter like A or B, or the word **tape**. The *xx* variable specifies the drive size (360, 720 1200, 1400).

/exattr=*srh* Exclude system, read-only, and hidden files from being backed up.

/addr=*base-i-d* Specifies the correct I/O address for certain tape controller cards where *base* is the hexadecimal base address of the card. The *i* is the interrupt channel (IRQ), and *d* is the DMA channel.

/rate=*x* Sets the data rate for specific tape controller cards that can support data transfer rates up to 1,000 kilobits per seconds (Kbps). Valid rates are: 1 for 1000Kbps, 5 for 500 Kbps, or 2 for 250 Kbps.

/full Specifies a full backup (the default setting).

/copy Specifies a full-copy backup method.

/fullerase Specifies a full backup to tape after first erasing the tape.

/inc Incremental backup (append to full).

/sep Separate incremental backup.

/dif Specifies differential backup.

/ecc Turns on error correction protocol.

/noecc Turns off error correction protocol.

/sf Formats diskettes with DOS formatting.

/nonsf Formats diskettes with WNBACKUP formatting.

/st Format tapes with QIC formatting.

/nonst Format tapes with WNBACKUP tape formatting.

/save Writes the history file to the hard disk and to the backup media.

/nosave Writes the history file to the backup media only.

/no Turns off the use of simultaneous hard disk and diskette DMA. Use this switch only if your computer locks up, your backup does not compare, or if you get a

```
General Hardware
Failure
```

message.

/dcheck Turns on the drive integrity check.

/nodcheck Disables the drive integrity check.

/i Starts WNBACKUP in minimized mode.

/b Starts WNBACKUP in backup mode and immediately prompts you to insert a diskette or tape.

/c Starts WNBACKUP in compare mode and immediately prompts you to insert the last disk or tape of the backup set you want to compare. The directory is read and the comparison automatically begins, using the specifications in the WNUSER.INI file.

/r Starts backup in the restore mode and immediately prompts you to insert the last diskette (or tape) of the backup set you want to restore.

/r *setup file name* Starts backup in restore mode and loads the settings that are saved in the specified setup file. This includes information such as the current drive and all files listed in the Include/Exclude Files dialog. To take effect, this switch must be specified before any others.

Notes

Windows must be in standard or enhanced mode to run WNBACKUP.

WNSCHEDL

WNSCHEDL is a Windows scheduler program. It enables you to specify a time and date for WNBACKUP to be started.

Type

Windows

Syntax

From the DOS command-line prompt:

win wnschedl

From the Windows desktop: Double-click on the Scheduler Icon in the IBM Tools folder.

Notes

Windows must be in standard or enhanced mode to run WNSCHEDL.

When WNSCHEDL is started, the Scheduler window is displayed.

WNUNDEL

Starts the Windows undelete program. The Undelete program recovers files by using information saved by delete protection methods. If no protection method was used, Undelete uses information available through DOS.

Type

Windows

Syntax

From the DOS command-line prompt:

win wnundel

From the Windows desktop: Double-click on the Undelete Icon in the IBM Tools folder.

Notes

Windows must be in standard or enhanced mode to run WNUNDEL.

WPCMINFO.CPL

WPCMINFO.CPL is Windows 3.1 icon that hooks into the Control Panel and displays information about PCMCIA sockets and cards on the host system. During the installation of this version of DOS, if you select PCMCIA support, then the WPCMINFO.CPL and WPCMINFO.HLP files are copied in your DOS directory. To install the information utility, copy these files into the WINDOWS\SYSTEM directory if Windows 3.1 is installed or if you install Windows 3.1 after installing this version of DOS. A PCMCIA icon is created in the Control Panel under Windows. Clicking on this icon accesses the PCMCIA support information utility.

XCOPY

Copies files (except hidden and system files) and directories, including subdirectories.

With this command, you can copy all the files in a directory, including the files in the subdirectories of that directory.

Type

DOS, External, Network

Syntax

xcopy *source* [*destination*][**/a**|**/m**][**/d:***date*][**/p**][**/s**[**/e**]][**/v**][**/w**]

Parameters

source Specifies the location and names of the files you want to copy. The *source* must include either a drive or a path.

destination
Specifies the destination of the files you want to copy. *Destination* can include a drive letter and colon, a directory name, a file name, or a combination.

Switches

/a Copies only source files that have their archive file attribute set. This switch does not modify the archive file attribute of the source file. For information about how to set the archive file attribute, see command "ATTRIB" on page 26.

/m Copies source files that have their archive file attributes set. Unlike the **/a** switch, the **/m** switch turns off archive file attribute in the files specified in *source*. For information about how to set the archive file attribute, see command "ATTRIB" on page 26.

/d:*date*
Copies only source files modified on or after the specified date. Note that the format of *date* depends on the COUNTRY setting you are using.

/p Prompts you to confirm whether you want to create each destination file.

/s Copies directories and subdirectories, unless they are empty. If you omit this switch, XCOPY works within a single directory.

/e Copies any subdirectories, even if they are empty. You must use the **/s** switch with this switch.

/v Verifies each file as it is written to the destination file to make sure that the destination files are identical to the source files.

/w Displays the following message and waits for your response before starting to copy files.

```
Press any key to begin copying file(s)
```

Notes

Default value for *destination*

If you omit *destination*, the XCOPY command copies the files to the current directory.

Specifying whether *destination* is a file or directory

If *destination* does not contain an existing directory and does not end with a backslash (\), XCOPY prompts you with a message in the following format:

```
Does DESTINATION specify a file name
or directory name on the target
(F = file, D = directory)?
```

Press **F** if you want the file(s) to be copied to a file. Press **D** if you want the file(s) to be copied to a directory.

XCOPY does not copy hidden and system files

To remove the hidden or system attribute from a file, use the ATTRIB command.

XCOPY sets archive attribute for destination files

XCOPY creates files with the archive attribute set, whether or not this attribute was set in the source file. For more information about file attributes, see command "ATTRIB" on page 26.

XCOPY vs. DISKCOPY

If you have a disk that contains files in subdirectories and you want to copy it to a disk that has a different format, you should use the XCOPY command instead of DISKCOPY. Since the DISKCOPY command copies disks track by track, it requires that your source and destination disks have the same format. XCOPY has no such requirement. In general, use XCOPY unless you need a complete disk image copy. However, XCOPY will not copy hidden or system files such as IBMBIO.COM and IBMDOS.COM. Therefore, use DISKCOPY to make copies of system disks.

XCOPY exit codes

The following list shows each exit code and a brief description of its meaning:

0 Files were copied without error.

1 No files were found to copy.

2 The user pressed **Ctrl+C** to terminate XCOPY.

4 Initialization error occurred. There is not enough memory or disk space, or you entered an invalid drive name or invalid syntax on the command line.

5 Disk write error occurred.

You can use the *errorlevel* parameter on the IF command line in a batch program to process exit codes returned by XCOPY. See the following "Examples" section.

Examples

The following example copies all the files and subdirectories (including any empty subdirectories) from the diskette in drive A to the diskette in drive B:

```
xcopy a: b: /s /e
```

The following example uses the */d:* and */v* switches:

```
xcopy a: b: /d:04/11/90 /v
```

In this example, only files on the diskette in drive A that were written on or after 04/11/90 are copied to the diskette in drive B. Once the files are written to the diskette in drive B, the XCOPY command compares the files on the two diskettes to make sure they are the same.

You can create a batch program to perform XCOPY operations and use the batch IF command to process the exit code in case an error occurs. For example, the following batch program uses replaceable parameters for the XCOPY *source* and *destination* parameters:

```
@echo off
rem COPYIT.BAT transfers all source
rem files in all directories on the source
rem drive (%1) to the destination drive (%2)

xcopy %1 %2 /s /e

if errorlevel 4 goto lowmemory
if errorlevel 2 goto abort
if errorlevel 0 goto exit

:lowmemory
echo Insufficient memory to copy files or
echo invalid drive or command-line syntax.
goto exit

:abort
echo You pressed Ctrl+C to end the xcopy operation.
goto exit

:exit
```

To use this batch program to copy all files in the C:\PRGMCODE directory and its subdirectories to drive B, type the following command:

```
   copyit c:\prgmcode b:
```

The command interpreter substitutes C:\PRGMCODE for *%1* and B: for *%2*, then uses XCOPY with the */e* and */s* switches. If XCOPY encounters an error, the batch program reads the exit code and goes to the label indicated in the appropriate IF *errorlevel* statement. DOS displays the appropriate message and exits from the batch program.

Related Information

For information about copying individual files, see commands "ATTRIB" on page 26, "COPY" on page 47, and "DISKCOPY" on page 80.

Chapter 5. Working With Device Drivers

This chapter describes the following installable device drivers supplied with this version of DOS. **Warning:** Use these advanced features only if you are an experienced DOS user and fully understand device drivers and memory management concepts.

Driver	Purpose
ANSI.SYS	Supports American National Standards Institute (ANSI) terminal emulation.
CMOSCLK.SYS	Supports the current date and time. It replaces the DOS system clock.
DISPLAY.SYS	Supports code page switching for monitors.
DRIVER.SYS	Creates a logical drive that you can use to refer to a physical disk driver and specifies parameters for a drive not supported by the ROM BIOS of your computer.
EGA.SYS	Saves and restores the display when a graphics program is used with an EGA display.
EMM386.EXE	On a computer with at least an 80386 microprocessor and extended memory, simulates expanded memory and provides access to the upper memory area.
HIMEM.SYS	Manages the use of extended memory on a computer with at least an 80286 microprocessor and extended memory. DOS Setup installs this device driver automatically on such systems.
INTERLNK.EXE	Performs serial and parallel communication and redirection of server drives and printers and makes it seem as if a remote computer is local.
PCMATA.SYS	Provides virtual block device driver support for PCMCIA ATA fixed disk cards if they are formatted with a FAT-compatible disk structure.
PCMCS.EXE	Loads PCMCIA Card Services support. Can be loaded either as a terminate-and-stay-resident program from the DOS command prompt or as a device driver when included in the CONFIG.SYS file.
PCMFDD.EXE	Provides diskette drive emulation on PCMCIA sockets as drives A and B. It can be loaded either as a terminate-and-stay-resident program from the DOS command prompt or as a device driver when included in the CONFIG.SYS file.
PCMMTD.EXE	Provides SHELL.MTD which demonstrates all of the interface capabilities in Memory Technology Drivers. It can be loaded either as terminate-and-stay-resident program from the DOS command prompt or as a device driver when included in the CONFIG.SYS file.
PCMSCD.EXE	Supports configuration of several PC cards. It can be loaded as a device driver in the CONFIG.SYS file, added as a command in the AUTOEXEC.BAT file, or loaded as a terminate-and-stay-resident program from the command line prompt.

PCMVCD.386 PCMVCD.386 is the Windows VxD (virtual device driver) for PC Card support in 386-enhanced mode. It is a replacement module for the Windows VCD (virtual COMM driver). PCMVCD.386 allows FAX and modem cards inserted into a PCMCIA socket to be available to all sessions under Windows.

PENDEV.SYS Provides the PENDOS application programming interface (API) for PENDOS applications.

POWER.EXE Provides the ability to reduce the consumption of power when your applications and devices are idle.

PRINTER.SYS Supports code-page switching for printers.

RAMDRIVE.SYS Simulates a hard disk drive by creating a virtual disk drive in the random access memory (RAM) of your computer.

RAMBOOST.EXE Increases the conventional memory of your computer (must be at least a 386 model) and makes using EMM386.EXE less complex. Automatically analyzes the current configuration of your computer and then reconfigures it automatically to load terminate-and stay-resident programs and device drivers into upper memory blocks.

SETVER.EXE Loads the DOS version table into memory.

SMARTDRV.EXE Loads the SMARTDRV.EXE device driver to perform double buffering. If you have a small computer system interface (SCSI) hard disk, you may need to use the double-buffering feature of SMARTDRV. Double-buffering provides compatibility for hard-disk controllers that cannot work with virtual memory.

UMBMONO.SYS Maps the video memory of the monochrome adapter as Upper Memory Blocks (UMBs) that are used for loading programs with the LOADHIGH command if a CGA, EGA, or VGA adapter is also present. It provides approximately 4K of extra memory.

UMBCGA.SYS Maps the video memory of a color adapter as Upper Memory Blocks (UMBs) that are used for loading programs with the LOADHIGH command if a plain monochrome adapter is also present. It provides approximately 16K of extra memory.

UMBHERC.SYS Maps the video memory of a Hercules** adapter as Upper Memory Blocks (UMBs) that are used for loading programs with the LOADHIGH command if a Hercules graphics adapter or a Hercules Graphics Adapter Plus is also present. It provides approximately 60K of extra memory.

UMBEMS.SYS Maps a 64K block of EMS memory as Upper Memory Blocks (UMBs) that are used for loading programs with the LOADHIGH command if an EMS card with at least 64K of EMS is also present in a machine It provides approximately 64K of extra memory. Works on any class of machine with any level of EMS memory driver, but the EMS 3.2 level drivers are preferred because it uses less memory when loaded.

ANSI.SYS

Defines functions that change display graphics, control cursor movement, and reassign keys. The ANSI.SYS device driver supports the use of ANSI escape sequences to control the screen and keyboard of your computer. An ANSI escape sequence is a sequence of ASCII characters, the first two of which are the escape character (1BH) and the left-bracket character (5BH). The character or characters following the escape and left-bracket characters specify an alphanumeric code that controls a keyboard or display function. Case is significant for all characters you use in ANSI escape sequences.

Syntax

device=[*drive*:][*path*] **ansi.sys** [**/x**][**/k**] [**/r**]

Parameters

[*drive*:][*path*]	Specifies the location of the ANSI.SYS file.

Switches

/x	Remaps extended keys independently on 101-key keyboards.
/k	Ignores extended keys on 101-key keyboards.
/r	Adjusts line scrolling to improve readability when ANSI.SYS is used with screen-reading programs, which make computers more accessible to people with disabilities.

Notes

Remapping extended keys:

If you have a keyboard with 101 keys, you might want to use the **/x** switch to remap certain extended keys. For example, there are two **Home** keys on keyboards with 101 keys. One **Home** key is on the numeric keypad and the other is in the block of cursor-control keys. To DOS, the two **Home** keys are the same, unless you specify the **/x** switch.

Ignoring extended keys:

Some computers do not reliably detect all the extended-keyboard services of 101-key keyboards. You can use the **/k** switch to force ANSI.SYS to ignore extended keys.

Note: You cannot use the **/k** and **/x** switches at the same time.

Parameters used in ANSI escape sequences:

Pn	Numeric parameter. Specifies a decimal number.

Ps Selective parameter. Specifies a decimal number that you use to select a function. You can specify more than one function by separating the parameters with semicolons.

PL Line parameter. Specifies a decimal number that represents one of the lines on your display or on another device.

Pc Column parameter. Specifies a decimal number that represents one of the columns on your screen or on another device.

ANSI escape sequences for cursor movement, graphics, and keyboard settings

In the following list of ANSI escape sequences, the abbreviation "ESC" represents the ASCII escape character 27 (1Bh), which appears at the beginning of each escape sequence.

ESC[*PL*;*Pc*H (Cursor Position.) Moves the cursor to the specified position (coordinates). If you do not specify a position, the cursor moves to the home position—the upper-left corner of the screen (line 0, column 0).

ESC[*PL*;*Pc*f (Cursor Position.) Moves the cursor to the specified position (coordinates). If you do not specify a position, the cursor moves to the home position—the upper-left corner of the screen (line 0, column 0).

ESC[*Pn*A (Cursor Up.) Moves the cursor up by the specified number of lines without changing columns. If the cursor is already on the top line, ANSI.SYS ignores this sequence.

ESC[*Pn*B (Cursor Down.) Moves the cursor down by the specified number of lines without changing columns. If the cursor is already on the bottom line, ANSI.SYS ignores this sequence.

ESC[*Pn*C (Cursor Forward.) Moves the cursor forward by the specified number of columns without changing lines. If the cursor is already in the column on the right, ANSI.SYS ignores this sequence.

ESC[*Pn*D (Cursor Backward.) Moves the cursor back by the specified number of columns without changing lines. If the cursor is already in the column on the left, ANSI.SYS ignores this sequence.

ESC[s (Save Cursor Position.) Saves the current cursor position. You can move the cursor to the saved cursor position by using the Restore Cursor Position sequence.

ESC[u (Restore Cursor Position.) Returns the cursor to the position stored by the Save Cursor Position sequence.

ESC[2J (Erase Display.) Clears the screen and moves the cursor to the home position (line 0, column 0).

ESC[K (Erase Line.) Clears all characters from the cursor position to the end of the line (including the character at the cursor position).

ESC[*Ps*;...;*Ps*m (Set Graphics Mode.) Calls the graphics functions specified by the following values. These specified functions remain active until the next occurrence of this escape sequence.

Text attributes:

0	All attributes off
1	Bold on
4	Underscore (on monochrome display adapter only)
5	Blink on
7	Reverse video on
8	Concealed on

Foreground colors:

30	Black
31	Red
32	Green
33	Yellow
34	Blue
35	Magenta
36	Cyan
37	White

Background colors:

40	Black
41	Red
42	Green
43	Yellow
44	Blue
45	Magenta
46	Cyan
47	White

Parameters 30 through 47 meet the International Standards Organization 6429 standard.

ESC[=*Ps*h (Set Mode.) Changes the screen width or type to the mode specified by one of the following values:

0	40x25 monochrome (text)
1	40x25 color (text)
2	80x25 monochrome (text)
3	80x25 color (text)
4	320x200 4-color (graphics)
5	320x200 monochrome (graphics)
6	640x200 monochrome (graphics)
7	Enables line wrapping
13	320x200 color (graphics)
14	640x200 color (16-color graphics)

15	640 x 350 monochrome (2-color graphics)
16	640 x 350 color (16-color graphics)
17	640 x 480 monochrome (2-color graphics)
18	640 x 480 color (16-color graphics)
19	320 x 200 color (256-color graphics)

ESC[=*Ps*l (Reset Mode.) Resets the mode by using the same values that Set Mode uses, except for 7, which disables line wrapping. The last character in this escape sequence is a lowercase L.

ESC[*code*;*string*;...p (Set Keyboard Strings.) Redefines a keyboard key to a specified string. The parameters for this escape sequence are defined as follows:

- *Code* is one or more of the values listed in the following table. These values represent keyboard keys and key combinations. When using these values in a command, you must type the semicolons shown in this table in addition to the semicolons required by the escape sequence. The codes in parentheses are not available on some keyboards. ANSI.SYS might not interpret some of the codes in parentheses unless you specify the **/x** switch in the DEVICE command for ANSI.SYS.
- *String* is either the ASCII code for a single character or a string contained in quotation marks. For example, both 65 and "A" can be used to represent an uppercase A.

Note: Some of the values in the following table are not valid for some computers. Check the documentation for your computer for values that are different.

ASCII Key Codes	*Code*	**SHIFT+***code*	**CTRL+***code*	**ALT+***code*
F1	0;59	0;84	0;94	0;104
F2	0;60	0;85	0;95	0;105
F3	0;61	0;86	0;96	0;106
F4	0;62	0;87	0;97	0;107
F5	0;63	0;88	0;98	0;108
F6	0;64	0;89	0;99	0;109
F7	0;65	0;90	0;100	0;110
F8	0;66	0;91	0;101	0;111
F9	0;67	0;92	0;102	0;112
F10	0;68	0;93	0;103	0;113
F11	0;133	0;135	0;137	0;139
F12	0;134	0;136	0;138	0;140
Home (numeric key)	0;71	55	0;119	—
Up Arrow (numeric key)	0;72	56	(0;141)	—

ASCII Key Codes	*Code*	SHIFT+*code*	CTRL+*code*	ALT+*code*
Page Up (numeric key)	0;73	57	0;132	—
Left Arrow (numeric key)	0;75	52	0;115	—
Right Arrow (numeric key)	0;77	54	0;116	—
End (numeric key)	0;79	49	0;117	—
Down Arrow (numeric key)	0;80	50	(0;145)	—
Page Down (numeric key)	0;81	51	0;118	—
Ins (numeric key)	0;82	48	(0;146)	—
Del(numeric key)	0;83	46	(0;147)	—
Home	(224;71)	(224;71)	(224;119)	(224;151)
Up Arrow	(224;72)	(224;72)	(224;141)	(224;152)
Page Up	(224;73)	(224;73)	(224;132)	(224;153)
Left Arrow	(224;75)	(224;75)	(224;115)	(224;155)
Right Arrow	(224;77)	(224;77)	(224;116)	(224;157)
End	(224;79)	(224;79)	(224;117)	(224;159)
Down Arrow	(224;80)	(224;80)	(224;145)	(224;154)
Page Down	(224;81)	(224;81)	(224;118)	(224;161)
Insert	(224;82)	(224;82)	(224;146)	(224;162)
Delete	(224;83)	(224;83)	(224;147)	(224;163)
Print Screen	—	—	0;114	—
Pause/Break	—	—	0;0	—
Backspace	8	8	127	(0)
Enter	13	—	10	(0,28)
Tab	9	0;15	(0;148)	(0;165)
Null	0;3	—	—	—
A	97	65	1	0;30
B	98	66	2	0;48
C	99	66	3	0;46
D	100	68	4	0;32
E	101	69	5	0;18
F	102	70	6	0;33
G	103	71	7	0;34
H	104	72	8	0;35
I	105	73	9	0;23
J	106	74	10	0;36
K	107	75	11	0;37
L	108	76	12	0;38

ASCII Key Codes	*Code*	SHIFT+*code*	CTRL+*code*	ALT+*code*
M	109	77	13	0;50
N	110	78	14	0;49
O	111	79	15	0;24
P	112	80	16	0;25
Q	113	81	17	0;16
R	114	82	18	0;19
S	115	83	19	0;31
T	116	84	20	0;20
U	117	85	21	0;22
V	118	86	22	0;47
W	119	87	23	0;17
X	120	88	24	0;45
Y	121	89	25	0;21
Z	122	90	26	0;44
1	49	33	—	0;120
2	50	64	0	0;121
3	51	35	—	0;122
4	52	36	—	0;123
5	53	37	—	0;124
6	54	94	30	0;125
7	55	38	—	0;126
8	56	42	—	0;126
9	57	40	—	0;127
0	48	41	—	0;129
-	45	95	31	0;130
=	61	43	—	0;131
[	91	123	27	0;26
]	93	125	29	0;27
\	92	124	28	0;43
;	59	58	—	0;39
'	39	34	—	0;40
,	44	60	—	0;51
.	46	62	—	0;52
/	47	63	—	0;53
'	96	126	—	(0;41)
ENTER (keypad)	13	—	10	(0;166)

ASCII Key Codes	*Code*	SHIFT+*code*	CTRL+*code*	ALT+*code*
/ (keypad)	47	47	(0;142)	(0;74)
* (keypad)	42	(0;144)	(0;78)	—
- (keypad)	45	45	(0;149)	(0;164)
+ (keypad)	43	43	(0;150)	(0;55)
5 (keypad)	(0;76)	53	(0;143)	—

Examples

To exchange the backslash and question-mark keys by using literal strings, type the following escape sequence:

```
ESC["\";"?"pESC["?";"\"p
```

To exchange the backslash and question-mark keys by using the ASCII value of each key, type the following escape sequence:

```
ESC[92;63pESC[63;92p
```

To restore the backslash and question-mark keys to their original values, type the following escape sequence:

```
ESC[92;92pESC[63;63p
```

CMOSCLK.SYS

Replaces the default DOS clock. If you install the CMOSCLK.SYS device driver, any request for the current date and time will access CMOSCLK.SYS instead of the DOS system clock.

Syntax

device=[*drive:*] [*path*] **cmosclk.sys**

Parameters

[*drive:***][***path***]**	Specifies the location of the CMOSCLK.SYS file.

Notes

Affect on other applications

Applications that use the DOS clock might not run properly or perform as efficiently using CMOSCLK.SYS.

When to install

Only install CMOSCLK.SYS if your current DOS system clock is not keeping the correct date.

Installing CMOSCLK.SYS

Use the following steps to install CMOSCLK.SYS:

1. Add the following statement to your CONFIG.SYS file:

```
device=c:\dos\cmosclk.sys
```

2. Restart your system by pressing **Ctrl+Alt+del**.

DISPLAY.SYS

Supports code page switching for your computer keyboard and display.

Syntax

device=[*drive*:][*path*] **display.sys** con[:]=(*type*[,[*hwcp*][,*n*]])

device=[*drive*:][*path*] **display.sys** con[:]=(*type*[,[*hwcp*][,(*n*,*m*)]])

Parameters

[*drive*:][*path*]	Specifies the location of the DISPLAY.SYS file.
type	Specifies the display adapter in use. The only valid value is **ega**. The **ega** value supports both EGA and VGA display adapters. If you omit the *type* parameter, DISPLAY.SYS checks the hardware to determine which display adapter is in use.
hwcp	Specifies the hardware code page. United States is the default.
n	Specifies the number of code pages in addition to the primary code page specified for the *hwcp* parameter. Valid values for *n* are in the range 0 through 6. This value depends on the number of code pages loaded at one time.
m	Specifies the number of subfonts the hardware supports for each code page. The default value is 2 if *type* is **ega**.

Notes

Using DISPLAY.SYS with monochrome or CGA display adapters

Because monochrome and CGA display adapters do not support code-page switching, using DISPLAY.SYS with either type of adapter has no effect.

Installing a third-party device driver

If you install both DISPLAY.SYS and a third-party device driver, such as VT52.SYS, the third-party device driver must be installed first. Otherwise, the third-party device driver might disable DISPLAY.SYS.

Examples

Suppose you want DISPLAY.SYS to support an EGA display adapter with a United States code page and the potential for two or more code pages without subfonts. To do this and to specify that DISPLAY.SYS is in the DOS directory on drive C, add the following line to your CONFIG.SYS file:

```
device=c:\dos\display.sys con:=(ega,437,2)
```

DRIVER.SYS

Creates a logical drive that you can use to refer to a physical diskette drive.

A logical drive is a pointer to a physical disk drive in your system. The logical drive is associated with a drive letter (for example, A or B). You can specify parameters to describe the disk drive to DOS.

Syntax

device=[*drive*:] [*path*]**driver.sys** /**d**:*number* [/**c**] [/**f**:*factor*] [/**h**:*heads*] [/**s**:*sectors*] [/**t**:*tracks*]

Parameters

[*drive*:][*path*]

Specifies the location of the DRIVER.SYS file.

Switches

/**d**:*number*

Specifies the number of the physical diskette drive. Valid values for *number* are in the range 0 through 127. The first physical diskette drive (drive A) is drive 0; a second physical diskette drive is drive 1; a third physical diskette drive, which must be external, is 2. For a computer with one diskette drive, both drives A and B are numbered 0; for a computer with multiple diskette drives, drive B is numbered 1.

/c

Specifies that the physical diskette drive can detect whether the drive door is closed (change-line support).

/**f**:*factor*

Specifies the type of diskette drive. Valid values for *factor* are as follows:

0	160K/180K or 320K/360K
1	1.2 megabytes (MB equals approximately 1 000 000 bytes)
2	720K (3.5-inch diskette) or other
7	1.44MB (3.5-inch diskette)
9	2.88MB (3.5-inch diskette)

The default value for *factor* is 2.

Generally, if you use the **/f** switch, you can omit the **/h**, **/s**, and **/t** switches. Check the default values for these switches to make sure they are correct for the type of diskette drive you are using. To determine the appropriate values for the diskette drive, see the documentation from the diskette drive manufacturer.

If you specify the **/h**, **/s**, and **/t** switches, you can omit the **/f** switch.

/h:*heads*
Specifies the number of heads in the diskette drive. Valid values for *heads* are in the range 1 through 99. The default value is 2. To determine the correct value for your diskette drive, see the documentation from the diskette drive manufacturer.

/s:*sectors*
Specifies the number of sectors per track. Valid values for *sectors* are in the range 1 through 99. The default value depends on the value of **/f:***factor*, as follows:

/s:9	**/f:0**
/s:15	**/f:1**
/s:9	**/f:2**
/s:18	**/f:7**
/s:36	**/f:9**

To determine the correct value for your diskette drive, see the documentation from the manufacturer of the diskette drive.

/t:*tracks*
Specifies the number of tracks per side on the block device. Valid values for *tracks* are in the range 1 through 999. The default value is 80, unless **/f:***factor* is 0, in which case the default value is 40. To determine the correct value for your diskette drive, see the documentation from the manufacturer of the diskette drive.

Notes

Disk-drive change-line support

The term *change-line support* means that a physical diskette drive can detect when the drive door is open and closed. Change-line support allows faster DOS operation with diskettes. If you specify the **/c** switch, it indicates to DOS that the physical disk drive can support change-line error detection. To determine whether your disk drive has change-line support, see the documentation from manufacturer of the disk drive. The **/f** switch is not required because the default is 2.

Modifying or redefining a supported physical disk drive

For information about modifying the parameters of a physical disk drive that is supported by your hardware, see command "DRIVPARM" on page 90. You can also use DRIVER.SYS to redefine a physical diskette drive.

Limitations on DRIVER.SYS

You cannot use DRIVER.SYS with hard disk drives. For information about substituting a logical drive letter for a hard disk drive, see the command “SUBST” on page 224.

Creating a duplicate logical drive

Suppose you want to use one physical diskette drive to copy files from one diskette to another. Because you cannot copy from and to the same logical drive by using the COPY or XCOPY command, you must assign a second drive letter to that physical drive.

If your system has just one physical diskette drive, you do not need to install DRIVER.SYS for this purpose. DOS already assigns both logical drive A and logical drive B to that drive. Just copy files from drive A to drive B and switch diskettes when DOS prompts you.

If your system has more than one diskette drive, then you need to use DRIVER.SYS to assign a second drive letter to the physical diskette drive.

Creating a new logical drive with different parameters

If you use DRIVER.SYS to assign a logical drive that has parameters different from those of the previously assigned logical drive, then the parameters of the previous logical drive will be invalid. Therefore, do not use the drive letter corresponding to the previous logical drive.

Examples

To add an external 720K drive to your system, add the following line to your CONFIG.SYS file:

```
device=driver.sys /d:2
```

Because no location is specified, DOS searches for DRIVER.SYS in the root directory of your startup drive.

Suppose you want to use a single 1.44-megabyte external disk drive to copy files from one diskette to another. To do this, you must add two identical DEVICE commands for DRIVER.SYS in your CONFIG.SYS file. This procedure assigns two logical drive letters to the same physical drive. You can then swap diskettes in the same drive during the copying process. The following example shows what needs to be added to the CONFIG.SYS file:

```
device=driver.sys /d:2 /f:7
device=driver.sys /d:2 /f:7
```

EGA.SYS

Saves and restores the display when the DOS Shell Task Swapper is used with EGA displays. If you have an EGA display, you must install the EGA.SYS device driver before using Task Swapper.

Syntax

device=[*drive*:][*path*]**ega.sys**

Parameters

[*drive*:][*path*]
Specifies the location of the EGA.SYS file.

Notes

If you are using a mouse on a system that has an EGA monitor, you can save memory by installing EGA.SYS before you install your mouse driver.

EMM386.EXE

Provides access to the upper memory area and uses extended memory to simulate expanded memory. This device driver must be loaded by a DEVICE command in your CONFIG.SYS file and can be used only on computers with an 80386 or higher processor.

EMM386 uses extended memory to simulate expanded memory for programs that can use expanded memory. EMM386 also makes it possible to load programs and device drivers into upper memory blocks (UMBs).

Syntax

device=[*drive*:][*path*]**emm386.exe** [**on**|**off**|**auto**][*memory*] [min=size][**w=on**|**w=off**]
[m*x*|frame=*address*|/p*mmmm*][pn=*address*][x=*mmmm-nnnn*][i=*mmmm-nnnn*] [b=*address*][l=minxms]
[a=altregs] [h=handles] [d=*nnn*][ram=*mmmm-nnnn*][noems] [novcpi] [highscan] [verbose]
[win=*mmmm-nnnn*][nohi] [rom=*mmmm-nnnn*][nomovexbda] [altboot]

Parameters

[*drive*:][*path*]
Specifies the location of the EMM386.EXE file.

[on | off | auto]
Activates the EMM386 device driver (if set to **on**), suspends the EMM386 device driver (if set to **off**), or places the EMM386 device driver in auto mode (if set to **auto**). Auto mode enables expanded-memory support and upper memory block support only when a program calls for it. The default value is ON. Use the EMM386 command to change this value after EMM386 has started.

memory
Specifies the maximum amount of extended memory (in kilobytes) that you want EMM386 to provide as expanded/Virtual Control Program Interface (EMS/VCPI) memory. This amount is in addition to the memory used for UMBs and EMM386 itself. Values for memory are in the range 64 through the lesser of either 32768 or the amount of extended memory available when EMM386 is loaded. The default value is the amount of free extended memory. If you specify the **noems** switch, the default value is 0. EMM386 rounds the value down to the nearest multiple of 16.

Switches

min=*size*

Specifies the minimum amount of EMS/VCPI memory (in kilobytes) that EMM386 will provide, if that amount of memory is available. EMM386 reserves this amount of extended memory for use as EMS/VCPI memory when EMM386 is loaded by the DEVICE=EMM386.EXE command in your CONFIG.SYS file. EMM386 might be able to provide additional EMS/VCPI memory (up to the amount specified by the MEMORY parameter) if sufficient XMS memory is available when a program requests EMS/VCPI memory. Values are in the range 0 through the value specified by the **memory** parameter. The default value is 256. If you specify the **noems** switch, the default value is 0. If the value of **min** is greater than the value of **memory**, EMM386 uses the value specified by **min**.

w=on | w=off

Enables or disables support for the Weitek coprocessor. The default setting is **w=off**.

m*x*

Specifies the address of the page frame. Valid values for *x* are in the range 1 through 14. The following list shows each value and its associated base address in hexadecimal format:

1 => C000h

2 => C400h

3 => C800h

4 => CC00h

5 => D000h

6 => D400h

7 => D800h

8 => DC00h

9 => E000h

10 => 8000h

11 => 8400h

12 => 8800h

13 => 8C00h

14 => 9000h

Only use values in the range 10 through 14 on computers that have 512K of memory.

frame=*address*

Specifies the page-frame segment base directly. To specify a specific segment-base address for the page frame, use the **frame** switch and specify the address you want. Valid values for address are in the ranges 8000h through 9000h and C000h through E000h, in increments of 400h. To provide expanded memory and disable the page frame, you can specify **frame=***none*; however, this can cause some programs that require expanded memory to work improperly.

/**p***mmmm*

Specifies the address of the page frame. Valid values for *mmmm* are in the ranges 8000h through 9000h and C000h through E000h, in increments of 400h.

p_n_**=**_address_

Specifies the segment address of a specific page, where _n_ is the number of the page you are specifying and address is the segment address you want. Valid values for _n_ are in the range 0 through 255. Valid values for address are in the ranges 8000h through 9C00h and C000h through EC00h, in increments of 400h. The addresses for pages 0 through 3 must be contiguous in order to maintain compatibility with version 3.2 of the Lotus/Intel/Microsoft Expanded Memory Specification (LIM EMS). If you use the **m**_x_ switch, the **frame** switch, or the **/p*mmmm*** switch, you cannot specify the addresses for pages 0 through 3 for the **/p*mmmm*** switch.

x=_mmmm-nnnn_

Prevents EMM386 from using a particular range of segment addresses for an EMS page or for UMBs. Valid values for _mmmm_ and _nnnn_ are in the range A000h through FFFFh and are rounded down to the nearest 4-kilobyte boundary. The **/x** switch takes precedence over the **/i** switch if the two ranges overlap.

i=_mmmm-nnnn_

Specifies a range of segment addresses to be used (included) for an EMS page or for UMBs. Valid values for _mmmm_ and _nnnn_ are in the range A000h through FFFFh and are rounded down to the nearest 4-kilobyte boundary. The **/x** switch takes precedence over the **/i** switch if the two ranges overlap.

b=_address_

Specifies the lowest segment address available for EMS banking (swapping of 16-kilobyte pages). Valid values are in the range 1000h through 4000h. The default value is 4000h.

l=minXMS

Ensures that the specified amount (in kilobytes) of extended memory will still be available after EMM386 is loaded. The default value is 0.

a=altregs

Specifies how many fast alternate register sets (used for multitasking) you want to allocate to EMM386. Valid values are in the range 0 through 254. The default value is 7. Every alternate register set adds about 200 bytes to the size in memory of EMM386.

h=handles

Specifies how many handles EMM386 can use. Valid values are in the range 2 through 255. The default value is 64.

d=_nnn_

Specifies how many kilobytes of memory should be reserved for buffered direct memory access (DMA). Discounting diskette DMA, this value reflects the largest DMA transfer that will occur while EMM386 is active. Valid values for _nnn_ are in the range 16 through 256. The default value is 16.

ram=_mmmm-nnnn_

Specifies a range of segment addresses to be used for UMBs and also enables EMS support. If you do not specify a range, EMM386 uses all available adapter space to create UMBs and a page frame for EMS.

noems

Provides access to the upper memory area but prevents access to expanded memory.

novcpi

Disables support for VCPI applications. This switch must be used with the **noems** switch. If you specify the **novcpi** switch without specifying the **noems** switch, EMM386 does not disable VCPI support. If you specify both switches, EMM386 disregards the **memory** parameter and the **min** switch. Disabling support for VCPI applications reduces the amount of extended memory allocated.

highscan

Specifies that EMM386 use an additional check to determine the availability of upper memory for use as UMBs or EMS windows. On some computers, specifying this switch can have no effect or cause EMM386 to identify upper memory areas as available when they are not. As a result, your computer might stop responding.

verbose

Directs EMM386 to display status and error messages while loading. By default, EMM386 displays messages only if it encounters an error condition. You can abbreviate **verbose** as **v**. To display status messages without adding the **verbose** switch, press and hold the **Alt** key while EMM386 starts and loads.

win=*mmmm-nnnn*

Reserves a specified range of segment addresses for the Microsoft Windows program instead of for EMM386. Valid values for *mmmm* and *nnnn* are in the range A000h through FFFFh and are rounded down to the nearest 4-kilobyte boundary. The **/x** switch takes precedence over the **/win** switch if the two ranges overlap. The **/win** switch takes precedence over the RAM, ROM, and **i** switches if their ranges overlap.

nohi

Prevents EMM386 from loading into the upper memory area. Normally, a portion of EMM386 is loaded into upper memory. Specifying this switch decreases available conventional memory and increases the upper memory area available for UMBs.

rom=*mmmm-nnnn*

Specifies a range of segment addresses that EMM386 uses for shadow RAM, which is random-access memory used for read-only memory (ROM). Valid values for *mmmm* and *nnnn* are in the range A000h through FFFFH and are rounded down to the nearest 4-kilobyte boundary. Specifying this switch can speed up your system if it does not already have shadow RAM.

nomovexbda

Prevents EMM386 from moving the extended BIOS data from conventional memory to upper memory.

altboot

Specifies that EMM386 use an alternate handler to restart your computer when you press **Ctrl+Alt+Del**. Use this switch only if your computer stops responding or exhibits other unusual behavior when EMM386 is loaded and you press **Ctrl+Alt+Del**.

Notes

Loading HIMEM.SYS before EMM386.EXE

You must include a DEVICE command for the HIMEM.SYS device driver in your CONFIG.SYS file before the DEVICE command for EMM386.EXE.

Using EMM386.EXE memory switches

Unless you want to use EMM386.EXE to provide access to the upper memory area, you need not specify memory switches on the DEVICE command line. EMM386.EXE usually runs properly with the default values. In some cases, however, you might want to control how EMM386.EXE uses memory. For example, some programs will run better if you allocate more expanded memory. Also, you can control where EMM386.EXE puts the EMS page frame, or which segments it uses for EMS pages. You can use as many of these memory switches as you want, in any order you want.

Warning: Use EMM386.EXE parameters carefully. You can disable your system if you use them incorrectly.

Using EMM386.EXE to provide access to the upper memory area

In addition to providing access to expanded memory, EMM386.EXE provides access to the upper memory area, which you can use to load certain programs and device drivers. You must use either the **ram** or **noems** switch to provide access to the upper memory area.

To give DOS access to the upper memory area but not to expanded memory, use the **noems** switch. To give DOS access to both the upper memory area and expanded memory, use the **ram** switch. The **ram** switch provides access to less of the upper memory area for running device drivers and programs than does the **noems** switch. In either case, you must include the DOS=UMB command in your CONFIG.SYS file. The device command for EMM386.EXE must precede any DEVICEHIGH commands.

If you are using a Virtual Control Program Interface (VCPI) application, such as Lotus 1-2-3 Version 3.1, use the **/ram** switch to provide access to expanded memory.

Inadequate space for page frame

If EMM386.EXE is unable to find 64K of contiguous space for the page frame, you will see the message:

```
Unable to set base address
```

Examples

To start EMM386 as an expanded-memory emulator, using the default values, add the following lines to your CONFIG.SYS file:

```
device=himem.sys
device=emm386.exe
```

Because no location is specified, DOS searches for EMM386.EXE in the root directory of your startup drive.

To allocate 4096K of memory to EMM386.EXE and specify that the EMM386.EXE file is located in the DOS directory on drive C, add the following line to your CONFIG.SYS file:

```
device=c:\dos\emm386.exe 4096
```

To emulate expanded memory, specify the segment-base address D000h for the EMS page frame, and allocate 512K of memory to EMM386.EXE, use one of the following commands:

```
device=emm386.exe 512 frame=d000

device=emm386.exe 512 p0=d000 p1=d400 p2=d800 p3=dc00
```

Suppose that, in addition to specifying the conditions set in the preceding commands, you want to prevent EMM386 from using the memory address E000h through EC00h. To do this and to specify that EMM386 can use 127 handles, add the following line to your CONFIG.SYS file:

```
device=emm386.exe 512 frame=d000 x=e000-ec00 h=127
```

To provide access to the upper memory area but not emulate expanded memory, add the following line to your CONFIG.SYS file:

```
device=emm386.exe noems
```

To provide access to the upper memory area and emulate expanded memory, add the following line to your CONFIG.SYS file:

```
device=emm386.exe ram
```

HIMEM.SYS

HIMEM is an extended-memory manager—a program that coordinates the use of the extended memory of your computer, including the high memory area (HMA) so that no two applications or device drivers use the same memory at the same time.

You install HIMEM by adding a DEVICE command for HIMEM.SYS to your CONFIG.SYS file. The HIMEM.SYS command line must come before any commands that start applications or device drivers that use extended memory. For example, the HIMEM.SYS command line must come before the EMM386.EXE command line.

Syntax

device=[*drive*:][*path*] **himem.sys**[**/a20control**:on|off] [**/cpuclock:on|off**][**/eisa**]
[**/hmamin=**_m_][**/int15=**_xxxx_] [**/numhandles=**_n_][**/machine:**_xxxx_] [**/shadowram:on|off**][**/verbose**]

In most cases, you do not need to specify command-line options. The default values for HIMEM.SYS are designed to work with most hardware.

Parameters

[*drive*:][*path*]

Specifies the location of the HIMEM.SYS file. HIMEM.SYS should always be located on the same drive that contains your DOS files. If the HIMEM.SYS file is in the root directory of your startup drive, you do not need to include a path. However, you must always include the complete filename (HIMEM.SYS).

Switches

/a20control:on|off

Specifies whether HIMEM is to take control of the A20 line even if A20 was on when HIMEM was loaded. The A20 handler gives your computer access to the HMA. If you specify /A20CONTROL:OFF, HIMEM takes control of the A20 line only if A20 was off when HIMEM was loaded. The default setting is **/a20control:on**

/cpuclock:on|off

Specifies whether HIMEM is to affect the clock speed of your computer. If the clock speed of your computer changes when you install HIMEM, specifying **/cpuclock:on** might correct the problem; however, enabling this option slows down HIMEM. The default setting is **/cpuclock:off**

/eisa

Specifies that HIMEM allocate all available extended memory. This switch is necessary only on an EISA (Extended Industry Standard Architecture) computer with more than 16MB of memory; on other computers, HIMEM automatically allocates all available extended memory.

/hmamin=*m*

Specifies the amount of kilobytes of memory an application must require for HIMEM to give that application use of the HMA. Only one application can use the HMA at a time; HIMEM allocates the HMA to the first application that meets the memory-use requirements set by this option. You can specify a value from 0 to 63.

Set **/hmamin** to the amount of memory required by the application that uses the most HMA memory.

The **/hmamin** option is not required; the default value is zero. Omitting this option (or setting it to zero) specifies that HIMEM allocate the HMA to the first application that requests it, regardless of how much of the HMA the application is going to use.

/int15=*xxxx*

Allocates the amount of extended memory (in kilobytes) to be reserved for the Interrupt 15h interface. Some older applications use the Interrupt 15h interface to allocate extended memory rather than using the XMS (extended-memory specification) method provided by HIMEM. If you use these applications, you can ensure enough memory is available to them by setting *xxxx* to 64K larger than the amount required by the application.

You can specify a value from 64 to 65535; however, you cannot specify more memory than your system has available. If you specify a value less than 64, the value becomes 0. The default value is 0.

/numhandles=*n*

Specifies the maximum number of extended-memory block (EMB) handles that can be used simultaneously. You can specify a value from 1 to 128; the default value is 32. Each additional handle requires an additional 6 bytes of memory.

/machine:*xxxx*

Specifies what type of computer you are using. Usually, HIMEM can detect your computer type successfully; however, there are a few computers that HIMEM cannot detect. On such systems, HIMEM uses the default system type (IBM AT or compatible). You might need to include the **machine** option if your computer is a type that HIMEM cannot detect and if HIMEM does not work properly on your system by using the default system type.

Currently, systems that require this option include Acer 1100, Wyse, and IBM 7552.

The value for *xxxx* can be any of the codes or their equivalent numbers listed in the following table.

Code	**Number**	**Computer type**
at	1	IBM AT or 100% compatible
ps2	2	IBM PS/2*
ptlcascade	3	Phoenix** Cascade BIOS
hpvectra	4	HP** Vectra (A & A+)
att6300plus	5	AT&T 6300 Plus
acer1100	6	Acer 1100
toshiba	7	Toshiba** 1600 & 1200XE
wyse	8	Wyse 12.5 Mhz 286
tulip	9	Tulip SX
zenith	10	Zenith ZBIOS
at1	11	IBM PC/AT (alternative delay)
at2	12	IBM PC/AT (alternative delay)
css	12	CSS** Labs
at3	13	IBM PC/AT (alternative delay)
philips	13	Philips**
fasthp	14	HP Vectra
ibm7552	15	IBM 7552 Industrial Computer
bullmicral	16	Bull Micral 60
dell	17	Dell** XBIOS

/shadowram:on|off

Specifies whether to disable shadow RAM (**/shadowram:off**) or to leave the ROM code running from RAM (**/shadowram:on**).
Some computers make ROM code run faster by shadowing it in

RAM—that is, by copying the ROM code into faster RAM memory at startup, which uses some extended memory. On computers that use shadow RAM and have less than 2MB of RAM, HIMEM usually attempts to disable shadow RAM to recover additional extended memory for programs to use. HIMEM can disable shadow RAM only on certain types of systems. When HIMEM disables shadow RAM, the ROM code runs in the slower ROM instead of RAM; therefore, your computer might run slightly slower than it did before.

/verbose

Directs HIMEM to display status and error messages while loading. By default, HIMEM does not display any messages unless it encounters an error. You can abbreviate **/verbose** as **/V**. (To display status messages without adding the **/verbose** switch, press and hold the **Alt** key while HIMEM starts and loads.)

Notes

Setting memory allocation

Only one program can use the high memory area at a time. If you omit the **/hmamin=***m* switch (or set it to 0), HIMEM.SYS reserves the HMA for the first program that requests it. HIMEM.SYS reserves the HMA for the first program that meets the memory requirements set by the **/hmamin=***m* switch. To ensure the most efficient use of your high memory area of your computer, set **/hmamin=***m* to the amount of memory required by the program that uses the most HMA memory.

Loading DOS into the high memory area

HIMEM.SYS or another XMS driver must be loaded before you can load DOS into the high memory area (HMA). You load DOS into the HMA by using the DOS=HIGH command in your CONFIG.SYS file.

Examples

To install HIMEM.SYS, using the default values, add the following line to your CONFIG.SYS file:

```
device=himem.sys
```

Since no location is specified, DOS searches for HIMEM.SYS in the root directory of your startup drive.

Suppose you want a program to use at least 40K of memory before it has access to the high memory area. To specify this and that HIMEM.SYS is located in the DOS directory of drive C, add the following line to your CONFIG.SYS file:

```
device=c:\dos\himem.sys /hmamin=40
```

To install HIMEM.SYS and specify the A20 handler for an IBM PS/2* computer, add either of the following lines to your CONFIG.SYS file:

```
device=himem.sys /machine:ps2
device=himem.sys /machine:2
```

In another scenario, you might want to install HIMEM.SYS and allow simultaneous use of as many as 128 extended-memory handles. Your HIMEM.SYS is located in the DEVICES directory on drive D. To do this, add the following line to your CONFIG.SYS file:

```
device=d:\devices\himem.sys /numhandles=128
```

INTERLNK.EXE

A file transfer utility program that lets you connect two computers using the serial ports or parallel ports through a client-server setup. The server computer runs the file transfer program. When a connection is made to the server computer, the client computer uses devices on the server computer as though they were local devices.

INTERLNK, the file transfer client, is a single device driver performing serial and parallel communication and redirection of server drives and printers. Use of the drives and printer attached to the remote computer makes the devices seem as though they are local.

Syntax

device=[drive:][path]**INTERLNK.EXE** [**/drives**:*n*] [**/noprinter**][**/com**[:][*n*|*address*]][**/lpt**[:][*n*|*address*]][**/auto**][**/noscan**][**/low**][**/baud**:*rate*] [**/v**]

Parameters

[drive:][path] Specifies the location of the INTERLNK.EXE file.

Switches

/drives:*n* Maximum number of drives to redirect. The default is 3.

/noprinter Specifies that printers not be redirected.

/com[:][*n*|*address***]** Specifies a serial port to use for data transfer. The N parameter specifies the number of the serial port. The address parameter specifies the address of the serial port. If you omit either parameter, the INTERLNK client searches all serial ports and uses the first port that it finds connected to the server. If you specify the **/com** switch and omit the **/lpt** switch, the client searches only for the serial ports. By default, INTERLNK scans all serial and parallel ports.

/lpt[:][*n"address***]** Specifies a parallel port to use for data transfer. The N parameter specifies the number of the parallel port. The address parameter specifies the address of the parallel port. If you omit either parameter, the INTERLNK client uses the first parallel port it finds connected to the server. If you specify the **/lpt** switch and omit the **/com** switch, the client searches only for the parallel ports.

/auto Does not load device driver if no connection is made at start up time.

/noscan Do not scan for connections when the system is restarted.

/low Load driver in conventional memory, even if upper memory is available.

/baud:*rate* Specifies the maximum baud rate for serial communication. Valid values of *rate* are: 9600, 19200, 38400, 57600, 115200 baud. The default baud rate is 115200.

/V Prevents conflicts with a computer's timer. Specify this switch if you have a serial connection between computers and one of them stops running when you use Interlnk to access a drive or printer port.

PCMATA.SYS

The PCMCIA virtual driver supports PCMCIA-ATA compatible fixed disks and SRAM cards formatted in a FAT structured format. This device driver, called PCMATA.SYS, registers as a bulk memory client to Card Services.

The PCMCIA virtual device driver also supports read/write operations for SRAM cards formatted in an DOS FAT structured format. You can also use the DOS FORMAT command to format SRAM PC Cards for read/write operations. Supporting this format also allows you to use data stored in other computers in the system with the PCMCIA Support software.

For access to the PCMCIA ATA IDE fixed disks, PCMATA.SYS talks directly to the IDE interface to read and write data. For PCMATA.SYS to function properly, all PCMCIA ATA IDE disks must be recognized and properly configured by Phoenix Card Services.

You can load PCMATA.SYS as a device driver in the CONFIG.SYS file. PCMATA.SYS must be loaded after Card Services.

Syntax

device=[*drive*:][*path*] **pcmata.sys /0|/1** [**/addr=***nn*]

Parameters

[*drive*:][*path*] Specifies the location of the device driver file.

Switches

/0|/1 Specifies the socket (0 or 1) to which drive emulation is assigned.

/addr=*nn* Specifies the system window base address of 8K for memory card emulation. The base address must be on a 16K boundary, where *nn* ranges from C0 to EE (C0 is equivalent to C000H, and EE is equivalent to EE00H).

Examples

As an example, to allocate a virtual drive to socket 0 in the system, add the following to your CONFIG.SYS file after the loading sequence of Socket Services:

```
device=[drive:][path]pcmcs.exe /addr=d0 /wait=12 /clients=10
device=[drive:][path]pcmscd
device=[drive:][path]pcmata.sys /addr=d4 /0
```

Once you load the virtual driver, you can use the following devices if they are formatted with a FAT-compatible diskette structure:

- PCMCIA ATA IDE fixed disks that have been properly configured at the alternate fixed disk address by Phoenix Card Services
- PCMCIA 1.0/2.0 compliant SRAM PC Card (read and write)
- PCMCIA 1.0/2.0 compliant One Time Programmable (OTP), Mask ROM, EEPROM

Using the DOS FORMAT Command to Format PCMCIA PC Cards:

PCMATA.SYS allows you to format PC Cards using the DOS FORMAT command. Use one of the following with the FORMAT command, where *drive* is the drive assigned to the socket.

For SRAM Cards:

For this size:	You would type:
256K	FORMAT *drive:* /U /T:64 /N:8
512K	FORMAT *drive:* /U /T:128 /N:8
1MB	FORMAT *drive:* /U /T:255 /N:8
2MB	FORMAT *drive:* /U

For IDE Fixed Disks, you use the same command. However, the parameters must correspond to the size of the fixed disk. The parameters are:

/U Performs an unconditional format.

/T:*n* Represents the number (*n*) of cylinders on the IDE Disk.

/N:17 Represents 17 sectors per track.

The following is an example using the FORMAT command to format an IDE Disk with 600 cylinders and 17 sectors per track:

```
format drive: /u /t:600 /n:17
```

For ATA rotating drive cards and SunDisk cards, use `format d: /u` to format the entire card.

PCMCS.EXE

PCMCS.EXE is a Card Services 2.0 driver that interfaces directly with Socket Services 2.0 which meets the Intel Exchangeable Card Architecture (ExCA).

PCMCS.EXE is responsible for coordinating access to the PC Cards and allocating system resources among Card Services client drivers. A *client* driver is a device driver, utility, or program designed to support a particular, or multiple PC Cards. Card Services 2.0 is only provided as a driver loaded by DOS. This driver can be loaded by the CONFIG.SYS file, run as a terminate-and-stay-resident (TSR) program from the DOS command prompt, or added as a command in your AUTOEXEC.BAT file. You must load Socket Services before you load Card Services.

Syntax

When loaded from the CONFIG.SYS file, include the following statement in your CONFIG.SYS file:

device=[*drive*:][*path*]**pcmcs.exe [/wait=***n***][/addr=***xx***][/irq=***n***][/clients=***n***] [/pmoff] [/regions][/?]**

Note: Socket Services is not installed with DOS. This software is provided with your system. If an error occurs while attempting to load Card Services, refer to the README.TXT file in your DOS directory for information about Socket Services.

Parameters

[*drive*:][*path*]
Specifies the location of the device driver file.

Switches

/wait=*n* Specifies the PCMCIA card insert settle time. This time is the delay time needed from the time a card is inserted in a socket to the time that Card Services can access it.

The value of *n* is the number of system timer ticks (18.2 per second) to wait. The default is 12. Some cards require a longer settle time than others.

/addr=*xx*
Specifies the starting segment address for Card Services for PCMCIA card configuration.

The value for *xx* is Start address. You must locate the segment (Start Address) within the first 1MB of address space and must specify the start address of a 2-digit hexadecimal segment address.

The granularity of the start address is dependent on the socket controller. However, the minimum address granularity allowed is 4K. If *addr* is not specified, PCMCS defaults to the first available 4K block starting at hex C0. The range for *xx* is hex C0 to F0 (where C0 is equivalent to hex C000, and hex F0 is equivalent to hex F000).

/irq=*n* Specifies the IRQ resource that is used by Card Services for PCMCIA card events. The value of *n* must be in the range of 8 to 15. If this switch is not specified, the default is 10.

/clients=*n*

Controls the maximum number of client drivers that can be registered with Card Services. Each client driver requires Card Services to allocate 60 bytes of memory. If this switch is not specified, the default is 10.

/pmoff This option controls the power management provided with Card Services. If this option is specified, PCMCIA cards are unilaterally powered-down on SUSPEND messages, and on RESUME messages the cards are powered-up. The cards are also reconfigured (if there is an associated client driver) by sending an artificial insertion message. When the system resumes, the cards must be reconfigured.

/REGIONS=*n*

Defines the maximum number of concurrent regions that Card Services will manage. A *region* corresponds to MTDs and is added to reduce the amount of memory required for Card Services.

/? This option displays a help message regarding command line parameters and their syntax.

Notes

Determining the default Resource Map

Card Services is responsible for building a map of resources available to PCMCIA cards. When PCMCS initializes, all standard devices are examined and the resources for these devices are reserved to avoid conflicting with each other. The variables are for reserving resources used by other devices that your computer might have (such as I/O, memory, and IRQs) that are not considered standard in an AT-compatible system.

The following list of devices is examined by Card Services when it is initialized:

- COM devices (COM1-COM4)
- LPT devices (LPT1-LPT3)
- Hard disk controllers (primary and secondary)
- Diskette controllers (primary and secondary)
- VGA adapters
- EGA adapters
- CGA adapter
- Monochrome adapter
- Game adapter (joystick)

PCMFDD.EXE

PCMFDD provides diskette drive emulation on PCMCIA sockets as drives A and B. PCMFDD provides INT13H emulation for drives A and B. When this driver loads, it performs diskette drive emulation on the specified socket. This driver registers with Card Services as a bulk memory client and is a completely hardware independent component of PCMCIA Support software.

After you load DOS, you can achieve diskette drive emulation by loading PCMFDD.EXE. You can load PCMFDD.EXE as a device driver from the CONFIG.SYS file. You can also load this device driver from the DOS command line as a TSR or add this command in the AUTOEXEC.BAT file. PCMFDD must be loaded after Card Services.

Add the following device driver to your CONFIG.SYS file:

```
device=[drive:][path]pcmfdd.exe /addr=nn /x:m
```

From the DOS command prompt, type:

```
[drive:][path]pcmfdd /addr=nn /x:m
```

Parameters

[*drive:*] [*path*] Specifies the location of the device driver file.

Switches

/addr=*nn* Specifies the system window base address of 8K for memory card emulation. The base address must be on a 16K boundary, where *nn* ranges from C0 to EE (C0 is equivalent to C000H, and EE is equivalent to EE00H)

x The emulated drive letter (drive A or drive B).

m The socket number (0 or 1).

Notes

For example, to configure socket 0 as drive A, add the following to the CONFIG.SYS after the loading sequence of Socket Services:

```
device=[drive:][path]pcmcs.exe /clients=11
device=[drive:][path]pcmfdd.exe /a:0
```

PCMMTD.EXE

PCMMTD provides SHELL.MTD, which demonstrates all of the interface capabilities in the MTD.

PCMMTD.EXE can be loaded as a terminate-and-stay-resident program from the DOS command prompt, add as a command in the AUTOEXEC.BAT file, or as a device driver from the CONFIG.SYS file. The PCMCS.EXE must be loaded prior to loading PCMMTD.EXE.

Syntax

Add the following device driver to your CONFIG.SYS file to install the Memory Technology Driver:

```
device=[drive:][path]pcmmtd.exe
```

From the DOS command prompt, type:

```
[drive:][path]pcmmtd.exe
```

Parameters

[*drive*] [*path*] Specifies the location of the device driver file.

PCMSCD.EXE

The PCMSCD.EXE is a super client driver that supports the configuration of several PC Cards. After configuration, the PC Card operates as an integral component of the system.

The PCMCIA Super Client Driver is intended to work only with the Phoenix PCMCIA Card Services and does not operate on Card Services furnished by third party suppliers.

You can load the Super Client Driver as a terminate-and-stay-resident program from the DOS command prompt, add as a command in the AUTOEXEC.BAT file, or as a device driver from the CONFIG.SYS file. You must load the Card Services driver before the Super Client driver.

Syntax

When loaded from the CONFIG.SYS file, include the following statement in your CONFIG.SYS file:

device=[*drive*:][*path*]**pcmscd.exe** [**/beep**][**/rs**=*speed*][**/com**=*port*][**/cards**][/**?**]

Parameters

[*drive*:][*path*] Specifies the location of the device driver file.

Switches

/beep Audible configuration acknowledgement. The default value is for the beep to be turned off.

Note: You will hear the beep only for the PC Cards that PCMSCD.EXE supports. There is a single beep if the PC Card is configured or reinserted successfully. A two-beep tone is heard on the removal of a PC Card. A three-beep tone is heard if the card is not properly configured.

/com=*port* Number of the communication port, where *port* can be 1, 2, 3, or 4. The default is 4.

/rs=*speed* IBM Token-Ring Network speed, where *speed* is 4 or 16. The default value is 4.

/? Displays help information.

/cards Displays a list of supported cards.

PCMVCD.386

PCMVCD.386 is the Windows VxD (virtual device driver) for PC Card support in 386-enhanced mode. It is a replacement module for the Windows VCD (virtual COMM driver). PCMVCD.386 allows fax and modem cards inserted into a PCMCIA socket to be available to all sessions under Windows. During installation of DOS, if you select PCMCIA support, PCMVCD.386 gets copied in your DOS directory. However, if you have Windows and want Phoenix PCMCIA Support, you must manually edit the SYSTEM.INI file before you can have Phoenix PCMCIA Support under Windows.

Syntax

To replace the Windows Virtual COMM driver with the PCMCIA Windows VxD driver:

1. Copy the file PCMVCD.386 from the DOS directory to the \WINDOWS\SYSTEM directory.
2. Edit the \WINDOWS\SYSTEM.INI file as follows:
   a. Locate the section label [386Enh].

   b. Add the following line anywhere in that section:

      ```
      device=d:\windows\system\pcmvcd.386
      ```

      where *d:* refers to the drive letter.

   c. Locate the line that states:

      ```
      device=*vcd
      ```

      and either comment it out by adding semi-colons at the start of the line:

      ```
      ; device=*vcd
      ```

      or

      Remove the line entirely.

   d. For each COM port that you might want to use as a PCMCIA port, add the following line to the [386Enh] section:

      ```
      com#base=xxxx
      ```

      where *#* is the number of the COM port (1, 2, 3, or 4) and *xxxx* is the hexadecimal value of the standard location of the COM port (3F8, 2F8, 3E8, or 2E8, respectively).

   e. If you want a window to pop up to notify you when a card is removed or inserted, add the line:

      ```
      pcmcianotify=true
      ```

      If you do not want notification, leave this line out.

   f. Add the line:

      ```
      pcmciacom4=true
      ```

      to enable the PCMCIA functions of the driver.

g. If the line *COMVERIFYBASE=TRUE* is present in the file, you must remove it. This option cannot be set and Windows still handle PCMCIA ports properly.

h. If the line:

   ```
   EMMExclude=xxxx-yyyy
   ```

   is present, edit it as follows; otherwise add it:

   ```
   EMMExclude=C800-DFFF
   ```

   Other values might work but are dependent on your particular system.

i. Now add a new section to SYSTEM.INI, called:

   ```
   [PCMCIA]
   ```

j. Make sure this line appears after all other items in a section. Each section starts with a name in [brackets].

k. For each COM port that you want available as a PCMCIA port, add the line:

   ```
   COM0#=TRUE
   ```

   where *#* is the COM port number (1, 2, 3, or 4).

3.

4. Save the new edited SYSTEM.INI file.

SYSTEM.INI Example:

An example to make COM ports 3 and 4 available as PCMCIA serial ports. Your SYSTEM.INI file would end up looking like this (only those parts relevant to PCMCIA are shown):

```
[386Enh]
DEVICE=C:\WINDOWS\SYSTEM\PCMVCD.386
; DEVICE=*vcd  << commented out!
EMMExclude=C800-DFFF
COM3BASE=3E8
COM4BASE=2E8
PCMCIACOM4=TRUE
PCMCIANOTIFY=TRUE

⋮
[PCMCIA]
COM03=TRUE
COM04=TRUE
```

Notes

- You cannot designate a port number for a COM port that is already installed. For example, if your computer has an adapter card already configured for COM1, you cannot also designate it as a PCMCIA port.
- If there are other entries in the [386Enh] section for *COM#*BASE=, then leave them. They are not relevant as long as the port numbers are not the same as ports you want to configure as PCMCIA ports.

- You must also designate a COM port number to the PCMCS.EXE device driver, either as part of your CONFIG.SYS file or to be loaded from the command line as a TSR.

  Note: The COM port specified to PCMCS must match one of the COM#BASE=*xxxx* and COM0#=TRUE entries in SYSTEM.INI.

When you initialize PCMVCD.386 under Windows, it registers as a client to Card Services. When you install a FAX and modem device, PCMVCD ensures that the device is available to the entire Windows operating system when you successfully configure the fax and modem card.

PENDEV.SYS

Provides the PENDOS application programming interface for PENDOS applications.

Syntax

device=[*drive*:][*path*]**pendev.sys**

Parameters

[*drive*:][*path*]	Specifies the location of the device driver file.

For more information, see command "PENDOS" on page 184.

POWER.EXE

The POWER.EXE device driver provides the ability to reduce the consumption of power when your applications and devices are idle. It is automatically installed when the installation program detects power management functions installed, on your computer. This device driver conforms to the Advanced Power Management (APM) specification.

Syntax

device=[*drive*:][*path*]**power.exe** [adv[:max|reg|min]|std|off][/low]

Parameters

[*drive*:][*path*]	Specifies the location of the POWER.EXE file.
[adv[:max|reg|min]	Conserves power when applications and hardware devices are idle. Performance might be affected if an application is active instead of idle. Use **max** for maximum power conservation. Use **reg**, the default setting, to balance power conservation with application and device performance. Use **min** if the performance of an application or device is not satisfactory when you specify **max** or **reg**.

std Conserves power by using only the power-management hardware features of your computer.

off Turns off power management.

If your computer does not support the Advanced Power Management specification, only the **std** parameter changes to specify the following:

std Turns off power management.

Switches

/low Loads the POWER.EXE device driver into conventional memory, even if the upper memory area is available. The POWER.EXE is loaded into the upper memory area, by default. If sufficient upper memory is not available then POWER.EXE loads low.

Notes

Add the following command line to your CONFIG.SYS file to:

- Specify that POWER.EXE is located in the DOS directory of drive C.
- Be sure to use the default setting.
- Be sure to load POWER.EXE into the upper memory area (if available).

```
device=c:\dos\power.exe
```

After the device driver is loaded, POWER.EXE can run from the DOS command prompt or BATCH file to display or change the current power settings.

```
power [adv[:max|reg|min]
|std|off]
```

To display the current power setting, type the following:

```
power
```

To change the current power setting, type the following:

```
power adv:max
```

This changes the current power setting for maximum power conservation.

Note: When loading CMOSCLK.SYS, it is important to load it after loading POWER.EXE. If you load it before, the clock fails to increment on RESUME.

PRINTER.SYS

Supports code page switching for the parallel ports PRN, LPT1, LPT2, and LPT3.

Syntax

device=[*drive:*][*path*]**printer.sys** lpt*x*=(*type*[,[*hwcp*] [,*n*]])

Parameters

[*drive*:][*path*]
Specifies the location of the PRINTER.SYS file.

lpt*x*
Specifies the number of the parallel port for which you want to support code-page switching.

type
Specifies the printer in use. The following list shows valid values for *type* and the printers represented by each value:

4201	IBM Proprinters II and III Model 4201 IBM Proprinters II and III XL Model 4202
4208	IBM Proprinter X24E Model 4207 IBM Proprinter XL24E Model 4208
5202,4019	IBM printers supporting Personal Printer Data Streams (PPDS)
EPS	Epson printers

hwcp
Specifies the hardware code page. United States is the default. See command "COUNTRY" on page 51 for code pages of other countries.

n
Specifies the number of code pages in addition to the code page specified in the *hwcp* parameter.

Examples

The following command loads the PRINTER.SYS device driver for use with the IBM Proprinter X24E Model 4207, loads code page 850, and prepares PRINTER.SYS to support two additional code pages:

```
  device=c:\dos\printer.sys lpt1:=(4208,850,2)
```

RAMBOOST.EXE

RAMBOOST increases the available conventional memory of your computer and reduces the complexity of using the DOS memory manager EMM386.EXE. It analyzes your configuration and automatically configures resident programs and device drivers to load into upper memory blocks. With RAMBOOST, there is no need to manually edit your CONFIG.SYS or AUTOEXEC.BAT files. RAMBOOST keeps a data file called RAMBOOST.INI in which it keeps track of the programs and device drivers that need to be loaded into upper memory. Using this file, it automatically makes the best use of the available upper memory.

Syntax

device=[*drive*:][*path*]**ramboost.exe [active][complete][disable][learn][load] [mode][pif] [sync][track** [*filename*[/**d**]]] [/**live**]

Parameters

[*drive*:][*path*] Specifies the location of the device driver file.

Switches

active Forces RAMBOOST to remain fully active and re-optimizes memory the next your computer is restarted anytime it detects a change to a tracked file. Active mode is the default.

complete Initial program loading is complete. Causes RAMBOOST to ignore every change in AUTOEXEC.BAT after this command.

disable Disables RAMBOOST by preventing it from loading.

learn Forces RAMBOOST to enter learn mode the next time you restart your computer, even if no tracked file has changed.

load Installs RAMBOOST resident in memory.

mode Returns a value, indicating the current status of RAMBOOST as not resident (0), in active mode (1), or in learn mode (2).

pif Displays any of the network drivers, TSRs, DOS tables, and any other files RAMBOOST has loaded into upper memory blocks.

sync Updates the signatures for all tracked files in the RAMBOOST.INI file. Use this when you have made a change to a tracked file but do not want RAMBOOST to relearn and optimize your system.

track [*filename*[/d]] Displays the names and signatures of the files that RAMBOOST currently tracks. The *filename* variable specifies a device driver or TSR to add (or delete if **/d** is specified) to the list of tracked files. The files in the list are tracked for possible changes that would affect memory usage. AUTOEXEC.BAT, CONFIG.SYS and RAMBOOST.INI are always tracked.

Notes

The RAMBOOST Configuration File

This file, RAMBOOST.INI, is an ASCII text file that can be edited. It contains all the necessary parameters for RAMBOOST to optimize memory usage. The RAMBOOST.INI file is initialized by RAMSETUP.

No LOADHIGH and DEVICEHIGH statements

RAMBOOST does not use LOADHIGH and DEVICEHIGH statements to load resident programs and device drivers into UMBs. Instead, it retains information on which device drivers and TSRs to load high in the RAMBOOST.INI configuration file and places them there itself.

RAMDRIVE.SYS

Creates a RAM disk in the random access memory (RAM) of your computer to simulate a hard disk drive. RAM disks are much faster than hard disk disks because the information they contain is always loaded into memory. RAM disks are temporary data you place on a RAM disk is lost when you turn off your computer. You can set up as many RAM disks as you want, limited only by the amount of memory your computer has. To do this, add one RAMDRIVE.SYS line to your CONFIG.SYS file for each RAM disk.

Syntax

device=[*drive*:][*path*]**ramdrive.sys** [*DiskSize SectorSize*][**/e**|**/a**]

device=[*drive*:][*path*]**ramdrive.sys** [*DiskSize SectorSize NumEntries*][**/e**|**/a**]

Parameters

[*drive*:][*path*]
Specifies the location of the RAMDRIVE.SYS file.

DiskSize
Specifies the size (in kilobytes) of the RAM disk. Valid values for *DiskSize* are in the range 4 through 31744. The default value i s 64.

SectorSize
Specifies the disk sector size (in bytes). Valid values for *SectorSize* are 128, 256, and 512. The default value is 512. If you include a value for the *SectorSize* parameter, you must also include a value for the *DiskSize* parameter. Although you can change the *SectorSize* value, the default value is strongly recommended.

NumEntries
Specifies the number of files and directories you can create in the root directory of the RAM disk. Valid values for *NumEntries* are in the range 2 through 1024. The default value is 64. If you include a value for the *NumEntries* parameter, you must also include values for the *DiskSize* and *SectorSize* parameters.

Switches

/e
Creates the RAM disk in extended memory instead of in expanded or conventional memory.

/a
Creates the RAM disk in expanded memory instead of in extended or conventional memory.

Notes

Using the NumEntries parameter

RAMDRIVE.SYS rounds the number you specify up to the nearest sector boundary. If there is not enough memory to create the RAM disk as specified, RAMDRIVE.SYS attempts to create it with a limit of 16 directory entries. This can result in a RAM disk with a different limit from the one you specified.

Using conventional memory

Although specifying a memory type is optional, it is strongly recommended. If you omit both the **/e** and **/a** switches, RAMDRIVE.SYS uses the conventional memory of your computer. It is not a good idea to use conventional memory for a RAM disk, because this reduces available work space for programs. However, if you do not have extended memory, expanded memory, or a hard disk drive, you might want to use conventional memory for a RAM disk. Because a RAM disk can increase the speed of a diskette system significantly, it can be worth the loss of some conventional memory.

Using extended memory

If your system has extended memory installed (starting at the 1-megabyte boundary), you can use this extended memory for one or more RAM disks. For RAMDRIVE.SYS to use extended memory, you must first install HIMEM.SYS or another extended-memory manager that conforms to the Lotus/Intel/Microsoft/AST extended memory specification (XMS). In your CONFIG.SYS file, the DEVICE command that installs the XMS extended-memory manager must precede the commands that install the RAM disk.

Using expanded memory

For RAMDRIVE.SYS to use expanded memory, you must configure your system so that it provides expanded memory. In your CONFIG.SYS file, the DEVICE command that installs the expanded-memory manager (such as EMM386.EXE) must precede the DEVICE command that installs RAMDRIVE.SYS. The expanded-memory manager must conform to the Lotus/Intel/Microsoft expanded memory specification (LIM EMS).

Increasing the efficiency of a RAM disk

For the best results with a RAM disk, you can define a TEMP environment variable and set it to point to a subdirectory on the RAM disk.

Examples

To create a RAM disk in extended memory and allocate 64K (the default amount) of extended memory to RAMDRIVE.SYS, add the following line to your CONFIG.SYS file:

```
device=ramdrive.sys /e
```

Because no location is specified, DOS searches for RAMDRIVE.SYS in the root directory of your startup drive.

Suppose you want to install RAMDRIVE.SYS in expanded memory and allocate 4 MB (4096K) of expanded memory to the RAM disk. To do this and to specify that RAMDRIVE.SYS is located in the DOS directory on drive C, add the following line to your CONFIG.SYS file:

```
device=c:\dos\ramdrive.sys 4096 /a
```

SMARTDRV.EXE

Loads the SMARTDRV.EXE device driver to perform double buffering. If you have a small computer system interface (SCSI) hard disk, you may need to use the double-buffering feature of SMARTDRV. Double-buffering provides compatibility for hard-disk controllers that cannot work with virtual memory.

To use the double-buffering feature of SMARTDRV, the SMARTDRV.EXE device driver must be loaded by a DEVICE command in your CONFIG.SYS file.

Syntax

device=[*drive*:] [*path*] **smartdrv.exe/double_buffer**

Parameters

[*drive*:][*path*]	Specifies the location of the SMARTDRV.EXE file.
/double_buffer	Specifies that SMARTDRV perform double buffering.

Notes

Determining if you need to use double-buffering

When the SETUP program is unable to determine whether double buffering is required for your system, it adds the command to to your CONFIG.SYS file automatically. To determine whether or not you can remove the SMARTDRV command line from your CONFIG.SYS file, proceed as follows:

1. Make sure the SMARTDRV driver has been loaded using the CONFIG.SYS command line and that double buffering is enabled.
2. At the command prompt, type **smartdrv** and then press **Enter**. SMARTDRV displays information about your system.
3. Look at the column labeled “Buffering”. If every line in this column reads “no”, you can remove the DEVICE command for SMARTDRV from your CONFIG.SYS file.

SETVER.EXE

Loads the DOS version table into memory.

SETVER.EXE loads into memory the DOS version table, which lists names of programs and the number of the DOS version with which each program is designed to run. To display or modify the version table, use command “SETVER” on page 209.

Syntax

device=[*drive*:][*path*]**setver.exe**

Parameters

[*drive*:][*path*]	Specifies the location of the SETVER.EXE file.

Note: If "DOS=HIGH" is used, placing SETVER.EXE after the "DOS=HIGH" line in the CONFIG.SYS file reduces the low memory usage of SETVER to zero.

UMBMONO.SYS

Maps the video memory of the monochrome adapter as Upper Memory Blocks (UMBs) that are used for loading programs with the LOADHIGH command if a CGA, EGA, or VGA adapter is also present. It provides approximately 4K of extra memory. UMBMONO.SYS works on any class machine from the 8088s and up.

Syntax

device=[*drive*:][*path*]**umbmono.sys**

Parameters

[*drive*:][*path*]	Specifies the location of the device driver file.

Notes

Programs that directly write into monochrome adapter memory are likely to crash the machine. UMBMONO.SYS includes defense mechanisms but they are not foolproof. If you experience frequent system hangs, you are probably using a program incompatible with UMBMONO.SYS. Remove UMBMONO.SYS from your CONFIG.SYS file and see if the problems go away.

Some programs and device drivers do not tolerate being loaded "high." The only way to tell if your program or device driver is able to is to try it or contact the application vendor.

UMBMONO.SYS can be used in conjunction with HIMEM.SYS on 286 class machines. Current versions of HIMEM require that UMBMONO.SYS be loaded after HIMEM.SYS is loaded.

UMBMONO.SYS refuses to load if another loaded driver is providing UMB support (such as EMM386).

The only video modes that should be used with this device driver are the color video modes.

Examples

To start UMBMONO.SYS and load a device driver named FOO.SYS high on a 286 system, add the following lines to your CONFIG.SYS file:

```
device=c:\dos\himem.sys
device=c:\dos\umbmono.sys
dos=high,umb
device=c:\dos\foo.sys
```

UMBCGA.SYS

Maps the video memory of a color adapter as Upper Memory Blocks (UMBs) that are used for loading programs with the LOADHIGH command if a CGA, EGA, or VGA adapter is also present in combination with a monochrome adapter. It provides approximately 16K of extra memory. UMBCGA.SYS works on any class machine from the 8088s and up.

Syntax

device=[*drive*:][*path*]**umbcga.sys**

Parameters

[*drive*:][*path*]	Specifies the location of the device driver file.

Notes

Programs that directly write into color adapter memory are likely to crash the machine. UMBCGA.SYS includes defense mechanisms but they are not foolproof. If you experience frequent system hangs, you are probably using a program incompatible with UMBCGA.SYS. Remove UMBCGA.SYS from your CONFIG.SYS file and see if the problems go away.

Some programs and device drivers do not tolerate being loaded "high." The only way to tell if your program or device driver is able to is to try it or contact the application vendor.

UMBCGA.SYS can be used in conjunction with HIMEM.SYS on 286 class machines. Current versions of HIMEM require that UMBCGA.SYS be loaded after HIMEM.SYS is loaded.

UMBCGA.SYS refuses to load if another loaded driver is providing UMB support (such as EMM386).

The only video mode that should be used with this device driver is mode 7 (plain monochrome).

Examples

To start UMBCGA.SYS and load a device driver named FOO.SYS high on a 286 system, add the following lines to your CONFIG.SYS file:

```
device=c:\dos\himem.sys
device=c:\dos\umbcga.sys
dos=high,umb
device=c:\dos\foo.sys
```

UMBHERC.SYS

Maps the video memory of a Hercules adapter as Upper Memory Blocks (UMBs) that are used for loading programs with the LOADHIGH command if a Hercules graphics adapter or a Hercules Graphics Adapter Plus is also present. It provides approximately 60K of extra memory. UMBHERC.SYS works on any class machine from the 8088s and up.

Syntax

device=[*drive*:][*path*]**umbherc.sys** [**/x**]

Parameters

[*drive*:][*path*]	Specifies the location of the device driver file.

Switches

/x	Omits video memory DMA test during startup. **Note:** This switch should only be used after an adapter has successfully passed the test. Adapters that fail this test are not compatible with UMBHERC.SYS.

Notes

Programs that use the Hercules card in its graphics modes are incompatible with this device driver. If the Hercules card is one of the “RAMFONT” type, any program that uses the RAMFONT capability is also incompatible with this device driver. Any program that sets a Hercules graphics mode or uses the RAMFONT capability is guaranteed to crash the computer. UMBHERC.SYS includes defense mechanisms but they are not foolproof. If you experience frequent system hangs, you are probably using a program incompatible with UMBHERC.SYS. Remove UMBHERC.SYS from your CONFIG.SYS file and see if the problems go away.

Some programs and device drivers do not tolerate being loaded “high.” The only way to tell if your program or device driver is able to is to try it or contact the application vendor.

UMBHERC.SYS can be used in conjunction with HIMEM.SYS on 286 class machines. Current versions of HIMEM require that UMBHERC.SYS be loaded after HIMEM.SYS is loaded.

UMBHERC.SYS refuses to load if another loaded driver is providing UMB support (such as EMM386).

Good candidates for loading high with UMBHERC.SYS are KEYB, DOSKEY, and PRINT. The extra 60K is also enough to load “high” disk compression driver on 8088 and 286 class machines that cannot run EMM386 to obtain UMBs. Video memory is slow so choose what to load with care.

The only video mode that should be used with this device driver is mode 7 (plain monochrome).

Examples

To start UMBHERC.SYS and load a device driver named FOO.SYS high on a 286 system, add the following lines to your CONFIG.SYS file:

```
device=c:\dos\himem.sys
device=c:\dos\umbherc.sys
dos=high,umb
device=c:\dos\foo.sys
```

UMBEMS.SYS

Maps a 64K block of EMS memory as Upper Memory Blocks (UMBs) that are used for loading programs with the LOADHIGH command if an EMS card with at least 64K of EMS is also present in a machine. It provides approximately 64K of extra memory. It works on any class of machine with any level of EMS memory driver, but the EMS 3.2 level drivers are is preferred because they uses less memory when loaded.

Syntax

device=[*drive*:][*path*]**umbems.sys**

Parameters

[*drive*:][*path*] Specifies the location of the device driver file.

Notes

UMBEMS.SYS will refuse to load if prior device drivers or programs have allocated any EMS memory.

UMBEMS.SYS allocates all EMS memory when it loads. If you have a card with more than 64K of EMS memory, you must decide if this is the best use of the EMS memory for your purposes because none is left over for other programs.

Some programs and device drivers do not tolerate being loaded "high." The only way to tell if your program or device driver is able to is to try it or contact the application vendor.

UMBEMS.SYS can be used in conjunction with HIMEM.SYS on 286 class machines. Current versions of HIMEM require that UMBEMS.SYS be loaded after HIMEM.SYS is loaded.

UMBEMS.SYS refuses to load if another loaded driver is providing UMB support (such as EMM386).

Good candidates for loading high with UMBEMS.SYS are KEYB, DOSKey, and PRINT. The extra 64K is also enough to load "high" disk compression driver on 8088 and 286 class machines that cannot run EMM386 to obtain UMBs.

Examples

To start UMBEMS.SYS and load a device driver named FOO.SYS high on a 286 system, add the following lines to your CONFIG.SYS file:

```
device=c:\dos\himem.sys
device=c:\dos\umbems.sys
dos=high,umb
device=c:\dos\foo.sys
```

Chapter 6. Configuring Mouse and RAMBoost .INI Files

It is possible to edit the .INI files in your system. It is recommended that only someone technically knowledgeable do so. This appendix provides information that can be used to edit your MOUSE and RAMBoost .INI fiels.

Creating or Modifying the MOUSE.INI File

A MOUSE.INI file comes into existence by one of the following methods:

- You create one.
- The DOS Setup program creates one when it detects your computer has a liquid crystal display (LCD), with a mouse installed.
- The DOS Setup program detects an existing MOUSE.INI file and leaves it intact.

To edit or create a MOUSE.INI file, use the same editing program you use to edit the CONFIG.SYS or AUTOEXEC.BAT files. Be sure to save the edited MOUSE.INI file, and then restart your computer. The following is an example of the MOUSE.INI file that is created by the DOS Setup program.

```
[MOUSE]
[DOSPointer]
PointerSize=Large
PointerColor=Normal
Growth=On
Threshold=25
Delay=5
```

The following list of possible keywords and values are given to assist you when customizing or creating a MOUSE.INI file.

File Parameter	Possible Values
[mouse]	
MouseType=	Bus
	Serialn; n=port number
	Inportn; n=port number
	PS2
HorizontalSensitivity=n	0 - 100; 50=DEFAULT
VerticalSensitivity	0 - 100; 50=DEFAULT
DoubleThreshold=n	0 - 100; 50=DEFAULT
ActiveAccelerationProfile	1 - 4; 2=DEFAULT
InterruptRate=n	1 - 4; 1=DEFAULT
CursorDisplayDelay	0 - 255; 0=DEFAULT
HardwareCursorSupport=TRUE, FALSE	TRUE=DEFAULT

ForceDefaultCursor=ON, OFF	OFF=DEFAULT
PhoenixBIOS=TRUE, FALSE	FALSE=DEFAULT
RotationAngle=n (Ballpoint mouse)	0 - 359; 0=DEFAULT
[DOSPointer]	
PointerSize=Small, Medium, Large	Small=DEFAULT
PointerColor=Normal, Reverse, Transparent	Normal=DEFAULT
Growth=ON, OFF	OFF=DEFAULT
Threshold=n	1 - 100; 1=DEFAULT
Delay=n	1 - 100; 1=DEFAULT
[WindowsPointer]	
PointerSize=Small, Medium, Large	Small=DEFAULT
PointerColor=Normal, Reverse, Transparent	Normal=DEFAULT
Growth=ON, OFF	OFF=DEFAULT
Threshold=n	1 - 100; 1=DEFAULT
Delay=n	1 - 100; 1=DEFAULT
[AccelerationProfilen]	1 - 4; NO DEFAULTS
Label=name of curve n	16-byte or less string
Movement=ml m2 ... mk	1 - 127; NO DEFAULTS (mk is an integer)
Factor=l1 l2 ... lk	.1 - 16.0; NO DEFAULTS (lk is a floating point x.xx)

Note: The **Movement** and **Factor** parameters for **AccelerationProfilen** must be set with 1-to-1 sequence, with a maximum of 32 values per row. For example:

```
[AccelerationProfile1]
label=slow
Movement=  1     5     7     9    11    37    39    41    43
Factor= 1.00  1.25  1.50  1.75  2.00  2.25  2.50  2.75  3.00

[AccelerationProfile1]
label=Moderate
Movement=  1     9    12    15    18    21    24    27    30
Factor= 1.00  1.25  1.50  1.75  2.00  2.25  2.50  2.75  3.00
```

Note: The settings in the MOUSE.INI file are read by the mouse driver, only when it is loaded. After you edit the MOUSE.INI file, at the DOS command prompt, type:`mouse off`; then, to reload the mouse driver, type: `mouse`.

Modifying the RAMBOOST.INI File

The RAMBOOST configuration file is an editable ASCII text file. It contains all the necessary parameters for the RAMBoost program to manage your system. The profile is named RAMBOOST.INI. When the RAMBoost program needs to consult or edit this profile, it searches for it in the:

- Current directory
- Directory from which the RAMBoost program was executed
- DOS search PATH, (if one exists).

The RAMBOOST.INI file, which has eight sections. Each section begins with a section header and contains specific assignments relevant to the operation of RAMBoost program. The sections are:

- System
- PIF.Advice
- Completion Triggers
- Text
- Learn
- PIF
- Learn.PIF
- Learn.UMB

Each section may contain comments you add, assignments, and blank lines. Comment lines must have a semi-colon as the first character. Assignments take on the form:

```
<tag>=<value>
```

Neither the <tag>, nor the <value> are case-sensitive. The following table shows the notational conventions used.

This Symbol	Indicates that the Parameter is
[ ] — Square brackets	Optional
{ } — Braces	Required
< > — Angle brackets	Replacement place holder

The following discussions describe the different sections in detail.

[System]

```
backup=<method>
```

Advises RAMBOOST as to which backup method to use when the RAMBOOST.INI file is altered. The choices are:

Method	Description
none	Does not save a backup of the file.
standard	Names the new file RAMBOOST.INI and the old file RAMBOOST.OLD. This is the default.
numbered	Names the new file RAMBOOST.INI and the old file RAMBOOST.*nnn* where *nnn* is the lowest 3-digit number that does not currently exist.

```
backupextensions=<extension>
```

When a backup copy is made of the RAMBOOST.INI file, the three-letter extension specified by this option is the extension of the backup copy. For example, if the extension is specified as CP8 then RAMBOOST.INI becomes RAMBOOST.CP8.

```
limit=<segment>
```

RAMBOOST measures the size of programs and device drivers during Learn mode. The limit tag restricts RAMBOOST's measurement of a program or device driver's load-time memory requirements to a specific segment address in low DOS memory.

```
lowfiles=<number>
```

Specifies the minimum number of DOS system file handles to be maintained in low DOS memory for compatibility with Windows. This value is calculated by the Ramsetup program and is inserted into the RAMBOOST.INI file at setup time. The Ramsetup program uses whichever value is greater: its calculation or the LowFiles assignment in the RAMSETUP.INI file.

```
loadquery=<prompt>
```

Specifies the level of prompting that RAMBOOST provides when loading into memory. The default value is yesno.

Valid prompt levels are:

Prompt	**Description**
full	Causes RAMBOOST to display three prompts (Y/N/A) when loading. The user selects Y to have RAMBOOST load, N to not have RAMBOOST load, and A to always load and suppress all prompts until RAMSETUP.EXE is run again.
yesno	Causes RAMBOOST to display two prompts (Y/N) when loading. The user selects Y to have RAMBOOST load, N to not have RAMBOOST load. This is the default.
none	Causes RAMBOOST to display no prompts. You can still interrupt RAMBOOST's loading by holding down CTRL+ALT and both SHIFT keys prior to its loading.

```
mode=<mode>
```

Specifies the mode of operation RAMBOOST uses when loading into memory. The various modes are:

Option	**Description**
learn	Instructs RAMBOOST to install in Learn mode. This is a special mode of operation where RAMBOOST watches, rather than assists DOS in executing the CONFIG.SYS and AUTOEXEC.BAT files. When your computer is restarted, RAMBOOST writes what it has learned back to the configuration file, disables Learn mode, and restarts the system in its normal operational mode.
active	The normal mode of operation. When RAMBOOST is active, it is capable of performing its functions. The optimizer step during setup normally places RAMBOOST into Active mode.
enabled	Similar to Active mode, except it will not automatically revert to Learn mode when the configuration changes. Instead, you are is asked whether to revert to Learn mode, or proceed in Active mode. RAMBOOST provides UMB management only.

Option	Description
disabled	Prevents RAMBOOST from installing itself in memory. If a MODE command is not found in the [System] section of the RAMBOOST profile, then RAMBOOST is disabled.

```
optimizerui=<filename> [<arguments>]
```

```
ouiminmem=<kbytes>
```

The values for these options are set by RAMBOOST and should not be modified.

```
option=<option>
```

These options enable or disable a particular RAMBOOST feature. The options are:

Option	Description
hma	On 386 or 486 computers that are not running an XMS manager with HMA support, such as HIMEM.SYS, this option requests that RAMBOOST provide HMA support.
umb	By default, RAMBOOST manages upper memory blocks. Enable this feature to have DOS manage these blocks.
binddevices	This option is set by RAMBOOST when it discovers that a new device, such as a CD-ROM, has been added to the system.
display	Enables RAMBOOST Mode display when you start your computer so that when you restart your computer, you see RAMBOOST's current mode displayed in text on your screen.
noframe	Disables the EMS page frame on LIM 3.2, LIM 4.0, and EEMS boards and drivers. This allows the 64K-page frame and any additional pages to be mapped as UMBs for use by RAMBOOST. No EMS memory is addressable while this option is invoked.
nohighalloc	Prevents DOS allocations from Upper Memory Blocks (UMB) by UMB-loaded programs.
nosmartframe	Disables the RAMBOOST default of SMARTFRAME. Tells RAMBOOST not to use the EMS page frame to assist in loading programs into high memory. RAMBOOST can also implicitly disable SMARTFRAME if it determines that using SMARTFRAME is not safe.
nosmartdma	Disables the SmartDMA feature (default). SmartDMA automatically allocates a single sector in the low memory buffer for some computer and XT* DMA controllers.
nosmarthddma	Disables the Smart Hard Disk DMA feature (default).
quiet	Disables the DOS table relocation messages.
smartstack	Instructs RAMBOOST to correct a CTRL+ALT+DEL problem that might occur with DOS's EMM386.EXE driver, or other 386 control programs.
smartdma	Protects transfers to and from diskettes. By default this option is set ON if all of these conditions are true: RAMBOOST platform is 386, UMB, or EMS; CPU type is 386SX, 386DX, or 486; and Virtual DMA Spec. (VDS) API is not installed.
smarthddma	Use when you have a hard drive that uses DMA. This is only valid when the Option SMARTDMA is in effect. This option is needed only on 386 or 486 computers with SCSI or ESDI hard-drive controllers, which use DMA for hard-disk transfers (MFM or RLL hard-drive controllers use programmed I/O for hard-disk transfers).

Option	Description
smartframe	Enables the SmartFrame if possible (default). SmartFrame temporarily borrows the EMS page frame for loading programs and device drivers.
recover	Attempts to run the setup program when RAMBOOST fails to load.
test	Performs a complete memory test on all DOS= and UMB= memory ranges.
verbose	Enables full DOS table relocation messages, allowing you to see RAMBOOST"s status while it relocates the DOS tables.

```
reset=<reset_mode>
```

Specifies the mode by which RAMBOOST will reset the processor when required. The options are:

Mode	Description
cold	Performs a cold restart (default).
hard	Performs a hard restart by pulling the processor reset pin low.
warm	Performs a warm restart.
fast	Performs a restart using interrupt 19h.

```
resetdelay=<seconds>
```

Advises RAMBOOST to wait <seconds> number of seconds before resetting the computer. The default is 0 seconds (no delay). It might be necessary to use a non-zero value to allow some time for disk-caching programs with delayed write options to flush their buffers to disk before the processor is reset.

Note: RAMBOOST attempts to advise disk caches of an impending reset by issuing the DOS function call to reset (before any delay).

```
setup=<filename.ext>
```

Specifies an override of the name of the RAMSETUP program. It is used during startup, when RAMBOOST needs to execute its setup program in response to a system problem that requires configuration recovery, such as a corrupted CONFIG.SYS file.

```
smartdmabuffersize=<bytes>
```

Is used in conjunction with the SmartDMA and SmartHDDMA options. This option overrides the default DMA buffer size. It is activated when either SmartDMA or SmartHDDMA is enabled.

For more information on SmartDMA and SmartHDDMA, see the system section tab, Option=<smartdma> explanation.

```
terminatedelay=<seconds>
```

Specifies the number of seconds to wait for an answer to the query `Load RAMBOOST [Y/N]`, which is issued at startup. When that number of seconds expires, without an answer, RAMBOOST continues to load. If the TerminateDelay=0, then the query is not issued.

Note: Regardless of the TerminateDelay setting, you can always cause a terminate query to appear at startup by holding down CTRL, ALT and both SHIFT keys simultaneously while RAMBOOST is loading.

[PIF.Advice]

```
    <filename.ext>={low|high}[=<segment>]
    [,locked]
```

These assignments are PIF (Program Information) assignments. PIFs record the name, size of each program or device driver, and a HIGH or LOW destination. These PIFs partially override information found in matching PIFs in the [PIF.Learn] section or in Learn mode collected data. If you specify HIGH with no <segment> specification, the PIF modifies a partial PIF it matches, and becomes subject to optimization. If you specify <segment> in conjunction with HIGH, then RAMBOOST attempts to load the program or device driver at or above the specified <segment> address.

[CompletionTriggers]

```
shell=<filename.ext>[, size=<size>]
```

RAMBOOST builds a list of file names from 0 or more shell= assignments in this section when starting a Learn-mode pass. During this pass, RAMBOOST compares the file name associated with every program execution against each file name in the list. When a match occurs, RAMBOOST optionally compares the size of the file to the size recorded in the list, (if present) to verify the match. RAMBOOST terminates Learn mode on the first confirmed match, and immediately invokes RAMBOOST active.

[Text]: These assignments work in conjunction with the option=display from the [System] section. This allows you to configure the text that will appear in the RAMBOOST Mode Display Tag. These strings are truncated at 29 characters and will be displayed entirely in uppercase. The following examples show text-string assignments that RAMBOOST recognizes.

```
    learntext=<text>
```

Replaces default text for Learn mode indicator. The default indicator is Learn.

```
    activetext=<text>
```

Replaces default text for Active mode indicator. The default is Active.

[Learn]: RAMBOOST uses the [Learn] section of its profile to keep information that controls the Learn mode operations. The information kept here is not important to users. When RAMBOOST optimizes memory, it collects certain data about the system at that time and records it in this section. Should any part of this data change, RAMBOOST reverts from Active to Learn mode automatically. This data is inspected and compared to existing conditions when RAMBOOST is loaded.

The assignments are:

Assignment	Description
ASIG=timestamp	Time stamp on AUTOEXEC.BAT.
CSIG=timestamp	Time stamp on CONFIG.SYS.
PSIG=timestamp	Time stamp on RAMBOOST.INI.
DSIG=DOS info	DOS version and OEM name.
BSIG=d:	Letter of the drive you are restarting from.
FSIG=serial #	Serial # (DOS + 4) of the drive you are starting from.

Note: These tags are only modified under the control of RAMBOOST and are presented for explanatory purposes only.

RAMBOOST writes these signatures after it optimizes memory successfully. When starting in Active mode, RAMBOOST reads each signature and compares these values to the current values in the system. If any signature is different, RAMBOOST reverts to Learn mode automatically. During startup when MODE=ENABLED, RAMBOOST performs the same tests, however, if a change has occurred, RAMBOOST only recommends reverting to Learn mode to query you.

If a signature does not exist in the file, the corresponding check is not performed. There may be 0 or more xsig assignments. These assignment s are created and deleted by the RAMBOOST Track utility, which allows you to configure RAMBOOST to track extra configuration files.

```
option=<option>
```

RAMBOOST observes some [Learn] options. The options are:

Option	Description
keepinfo	When RAMBOOST is invoked at the end of a Learn mode, this option causes RAMBOOST to write its collected data to the file. Enables RAMBOOST to perform offline optimizations later, based on that data. RAMBOOST writes this data to the [Learn.PIF] and [Learn.UMB] sections, which are described further in this chapter.
noabortreset	Normally, RAMBOOST attempts to reset the computer when activated by the completion of Learn mode (online optimization). This option prevents RAMBOOST from resetting the computer when the process is terminated prematurely by operator intervention.

```
optdelay=<seconds>
```

Causes RAMBOOST to wait <seconds> number of seconds after completion of an optimization before exiting from or resetting the system. The default is 0.

```
optmethod=<method>
```

Advises RAMBOOST as to which optimization method to use. The <method> is a variable number that has the following meaning:

Number	Description
0	Requires RAMBOOST to choose either the detailed method (1) or the quick method (2), based on overall upper memory requirements, and the amount of upper memory available. This is the default.
1	Forces RAMBOOST to always perform a detailed (1-pass) optimization. This method is the most precise, however, it can be time-consuming sometimes.
2	Forces RAMBOOST to always perform a quick (2-pass) optimization. This method is not as precise as the detailed method, however it is generally much faster.
-1	Similar to method 0, except upon making its decision, RAMBOOST allows you to reverse its decision.

```
fit=<strategy>
```

This assignment alters the order in which RAMBOOST attempts to fit programs or device drivers into upper memory. The default is first. In general, this does not affect the operation of RAMBOOST. Three strategies are available:

Strategy	Description
FIRST	Use the first fit strategy when allocating Upper Memory Blocks. This option only works when RAMBOOST is in Passive mode.
BEST	Allocates from the smallest Upper Memory Block, which satisfies the size of an Upper Memory Block request. This option only works when RAMBOOST is in Passive mode.
LAST	Use the last fit strategy when allocating Upper Memory Blocks. This option only works when RAMBOOST is in Passive mode.

[PIF]

These are examples of PIF entries:

```
ramboost=      high=D000, need=10256, keep=10256
ramdrive.sys=  high=C800, need=6912,  keep=1136,  size=5873
setver.exe=    high=B000, need=14736, keep=400,   size=12007
winox.sys=     high=C800, need=4032,  keep=2784,  size=2998
mouse.sys=     low,       need=56032, keep-17040, size=55007
files=         high=D000, need=1600,  keep=1600
fcbs=          high=D000, need=256,   keep=256
buffers=       high=D000, need=512,   keep=512
drives=        high=D000, need=624,   keep=624
stacks=        high=C800, need=624,   keep=624
stacks=        high=C800, need=2384,  keep=2384
smartdrv.exe=  high=DA00, need=44688, keep=26896, size=43609
ipx.com=       high=B000, need=28752, keep=19824, size=27459
netx.com=      high=DA00, need=53728, keep=43872, size=52443
doskey.com=    high=C800, need=7888,  keep=5744,  size=5883
```

This section defines the memory requirements of specific programs and device drivers. Each assignment in this section has the general format:

```
filename.ext=data
```

where filename.ext is the name of the program or device driver image file. There must not be a path associated with filename.ext.

The data associated with the assignment can contain one or several of the following expressions. Individual expressions are separated by commas and at least one space.

Note: Do not change the order of entries in this section. RAMBOOST sequentially processes PIFs . If you change the order of these entries, they might not be matched correctly to their respective programs or device drivers.

Each PIF can have the following clauses in the assignment, each separated by a comma:

```
low or high[=<segment>]
```

LOW advises RAMBOOST not to relocate this program or device driver into upper memory. This is the same as having no PIF match a program or device driver being loaded. HIGH informs RAMBOOST that the program or device driver will be loaded into upper memory.

If <segment> is specified in conjunction with HIGH, then RAMBOOST attempts to load the program or device driver at or above the specified <segment> address.

```
need=<bytes>
```

Specifies the minimum amount of memory the program needs to load into memory and initialize.

```
keep=<bytes>
```

Informs RAMBOOST about how much initial memory the program or device driver requires, and how much it keeps when it is loaded resident. RAMBOOST allocates at least as much memory as the greater of these two values.

```
size=<bytes>
```

This field is optional. If present, RAMBOOST compares the value to the size of the file name that matched the PIF <tag>. This provides extra verification that this PIF is to be used with the current program or device driver.

```
ems
```

Instructs RAMBOOST to make the PIF available for SMARTFRAME loading. RAMBOOST normally adds this to the PIF command during the Learn process if it determines that the PIF would benefit from SMARTFRAME loading.

```
locked
```

Prevents RAMBOOST from altering a specified existing PIF record. A PIF that is locked low will always be loaded into low memory, while a PIF that is locked high will always be loaded into HIDOS UMB memory. When locking a PIF manually, remember that you must insert a comma and at least one space before the locked command.

For example:

```
files=low, locked
```

is the correct syntax. RAMBOOST generates the correct syntax if allowed to add the PIF on its own.

[Learn.PIF] These are examples of LEARN.PIF entries:

```
ramboost=      high,  need=10256,  keep=10256
ramdrive.sys=  high,  need=5888,   keep=1136,   size=5873
setver.exe=    high,  need=13712,  keep=400,    size=12007
winox.sys=     high,  need=3008,   keep=2784,   size=2998
mouse.sys=     high,  need=55008,  keep=17040,  size=55007
files=         high,  need=1600,   keep=1600
fcbs=          high,  need=256,    keep=256
buffers=       high,  need=512,    keep=512
drives=        high,  need=624,    keep=624
stacks=        high,  need=624,    keep=624
stacks=        high,  need=2384,   keep=2384
command.com=   high,  need=48416,  keep=5584,   size=47845
smartdrv.exe=  high,  need=43664,  keep=26896,  size=43609
ipx.com=       high,  need=27728,  keep=19824,  size=27459
netx.com=      high,  need=52704,  keep=43872,  size=52443
doskey.com=    high,  need=6864,   keep=5744,   size=5883
```

RAMBOOST generates this section when activated by the completion of a Learn-mode pass (online optimization), and when the [Learn] Option=KeepInfo is in effect. The assignments generated and placed in this section are Partial PIF (Partial Program Information) assignments. These PIFs record the name and size of each program or device driver RAMBOOST measured as a candidate for upper memory, along with its initial and final memory requirements.

[Learn.UMB]: These are examples of LEARN.UMB entries:

```
umb=C800-CBFF
umb=D000-D3FF
umb=B000-B7FF
umb=DA00-EFFF
umb=0000-9FFF
```

RAMBOOST generates this section when activated by the completion of a Learn-mode pass (online optimization), and when the [Learn] Option=KeepInfo is in effect. The assignment is:

```
umb=<range>
```

This specifies the base segment address and top segment address of an Upper Memory Block, which is available for use by the RAMBoost program when it starts. The DOS memory arena is also recorded in a UMB assignment.

Part 2: Error Messages

This part of the book contains an alphabetic listing of error messages associated with DOS, along with the cause of each message and the suggested action to take to resolve the problem. It contains the majority of error messages that are likely to be encountered. However, if an error message is displayed that is not shown in this book, chances are it was generated by an application. Refer to the documentation that came with the application for help.

Chapter 7. Messages

The messages in this book are the same as those displayed on your computer screen. A message could be for information only, or as the result of some problem that needs your intervention.

In this book, **bold** type indicates the message. The cause and action statements follow the message.

The first word in the cause of each message is the name of the program or command that generated the message. Sometimes, the message is generated by several different programs or commands. In this case, the first word is "COMMAND." When an internal DOS command generates the message, the first word is "DOS." If the command or program is not determined, only the cause is shown.

If the first part of the message contains a network number, NET###, refer to the *IBM PC Network Program* or the *IBM PC Local Area Network Program* for the cause of the message and the action to take.

Responses

When DOS displays device error messages, it also displays possible responses from which to choose. If you know what caused the problem, take corrective action before choosing one of the responses.

Following is a description of the possible responses. Make the responses in the following order until a response is received successfully.

R **Retry** the operation because the error might not occur again. The system tries the read or write operation again. We strongly recommend that you use this response first.

A **Abort** the program. The system ends the program that requested the read or write operation.

F **Fail** the current DOS system call and continue processing the program. Be careful when choosing this response if there is risk of damage to data.

I **Ignore** the error condition and continue the program. (Be careful when choosing this response because you may lose data.) The system is unable to determine if the condition is unsafe.

When DOS detects an error during reading or writing to any of the devices (disk drives, printer, and so on) on your system, it displays a message in the following format:

```
<type> error reading <device>
Abort, Retry, Ignore, Fail?
```

or

```
<type> error writing <device>
Abort, Retry, Ignore, Fail?
```

In these messages, *<device>* is the name of the device in error, such as PRN or B:, and *<type>* is one of the types listed on the following pages. The options (Abort, Retry, Ignore, or Fail) are related to a specific error message displayed. A disk error may display the following message: `Abort, Retry, Fail?` The "Ignore" option is not displayed.

A

A bad UMB number has been specified

Cause: LOADHIGH. When using the LOADHIGH command with the /L parameter, you specified an invalid region in the upper memory block (UMB).

Action: You can use LOADHIGH without the /L parameter and LOADHIGH will find an available region, or you can specify a valid region. To determine which regions are valid, use the MEM /F command.

Access denied

Cause: COMMANDS. Processing the requested command violates the access mode of the file, subdirectory, or device.

For example:

- You attempted to write to a file marked read-only or to a write-protected disk.
- You attempted to open a subdirectory as a file.

Action: Use a different file name or, if the file is read-only and you need to use that file, change its attribute with the ATTRIB command.

Active code page not available from CON device

Cause: KEYB. The KEYB status function was requested, but KEYB could not determine the loaded CON code page. Either the code page switching CON driver has not been installed, or there is no currently loaded CON code page.

Action: Install code page switching CON driver if needed.

All available space in the Extended DOS Partition is assigned to logical drives

Cause: FDISK. All the space in the extended partition has been assigned to logical drives.

Action: Remove or reduce the size of the existing logical drives and run FDISK again to create a logical drive.

All logical drives deleted in the Extended DOS Partition

Cause: FDISK. There are no more logical drives defined in the extended DOS partition.

Action: Delete the extended DOS partition if you no longer need it on the hard disk.

Allocation error, size adjusted

Cause: CHKDSK. A file name precedes this message. The file size as indicated by the directory is different from the file size indicated by the number of clusters allocated to the file. The file was truncated to match the amount of data allocated.

Action: If you specified the /F parameter, the new file size is changed to the number of clusters in the file times the number of bytes per cluster. To correct the file size, type:

```
CHKDSK /F
```

An incompatible DOSKEY is already installed

Cause: DOSKEY. There is more than one version of the DOSKEY program on your computer, and the installed version is not compatible with your system.

Action: Locate all versions of the program, and delete all but the correct one. If you are not sure which version is correct, copy the correct version from your DOS installation diskette.

ANSI.SYS must be installed to perform requested function

Cause: MODE. The requested screen function requires the extended support of ANSI.SYS.

Action: Add DEVICE=ANSI.SYS to your CONFIG.SYS file and restart DOS.

APPEND already installed

Cause: APPEND. You have already used the APPEND command once since you turned on your computer. You are now trying to use either the /x or /e switch with the APPEND command. These switches are valid only the first time you type the APPEND command.

Action: To change the append switch, restart your computer. Then type APPEND with the desired switch. Otherwise, use the APPEND command without these switches.

APPEND / ASSIGN Conflict

Cause: APPEND. You cannot use the APPEND command on an assigned drive.

Action: Cancel the drive assignment before using the APPEND command with this drive again.

Attempt to write on write-protected diskette

Cause: COMMANDS. Either the file is read-only or there is not enough available free storage space left on the disk.

Action: Do one of the following:

- Specify a file that can be written to.
- Make available free storage space by erasing unneeded files.
- Use the ATTRIB command to change the attributes of the file you want to write to.

For information about changing file attributes, see "ATTRIB" on page 26.

B

/B invalid with a black and white printer

Cause: GRAPHICS. You tried to print background color on a black and white printer.

Action: Do not use the /B parameter.

Bad command or file name

Cause: COMMANDS. The command you entered is not a valid DOS command.

Action:

- Check the spelling of the command and re-enter it.
- If you spelled the command name correctly, make sure that the default drive contains the external command or batch file you are trying to process.

Bad command or parameters

Cause: IBMBIO. As DOS read and interpreted the commands within the CONFIG.SYS file, it identified a bad command or parameter.

Action: Edit the CONFIG.SYS file and correct the statement where the error occurred.

Bad disk in drive

Cause:

- The wrong disk or tape may have been inserted (For example, a 1.2MB disk was expected, but a 360K was inserted)
- Also, if you are using a drive other than A or B to back up to, and the disk is not formatted, this error message appears.
- Check your backup speed. If the inserted disk was used in a high- or medium-speed backup, and the current setting is low speed, this error may occur.

Action:

- Insert the correct disk or tape.
- Backup requires preformatted media when you are using the Fixed or Removable drive and path options, or if you are backing up to disks in drives other than A or B.
- In the third instance, select the Format button in the Disk Insert dialog box.

Bad drive-request structure length

Cause: QCONFIG. Hardware error has occurred.

Action: No action required.

Bad or missing *d:\path\devicedriver*

Cause: COMMANDS. The device driver file is not on the drive or the directory listed in the command line.

Action: Edit your CONFIG.SYS file and type the correct drive and path name in the device driver line.

Bad or missing Command Interpreter

Cause: DOS. This message suggests one of the following:

- The disk that you are using to start DOS does not contain a copy of COMMAND.COM, or an error occurred while DOS was loading the disk.
- COMMAND.COM has been removed from the directory it was in originally when you started DOS.
- The COMSPEC= parameter in the environment points to a directory not containing COMMAND.COM, and DOS is trying to reload the command processor.

Action: Restart DOS (press and hold Ctrl+Alt ; then press Del). If the system reset fails to solve the problem, start DOS with your backup DOS diskette, and copy COMMAND.COM from the backup diskette to the root directory of the disk that failed.

Bad or missing Keyboard Definition File

Cause: KEYB. The KEYBOARD.SYS file could not be found, contained invalid data, or is in the wrong subdirectory.

Action: Make sure that the specified KEYBOARD.SYS file exists and is in the appropriate subdirectory. If the file exists, get a new KEYBOARD.SYS file from the original DOS diskette.

Bad Partition Table

Cause: FORMAT. The partition table of the hard disk does not have a DOS partition or the partition table is invalid.

Action: Load FDISK, set up a new DOS partition on the hard disk, and retry the FORMAT command.

Batch file missing

Cause: DOS. DOS cannot find the batch file it was processing. The file might have been erased or renamed by one of the steps within it or the current drive might have been changed within the batch file, and the batch processor can no longer find the .BAT file used by your PATH. Batch processing stops and the command prompt appears.

Action:

- If the file name was changed, correct the command that changed the name.
- If the file was erased, use your backup copy. If you used EDLIN to create the file or make changes, rename the .BAK file to .BAT. Correct the command that deleted the file.
- Include the drive letters in the PATH.

Baud rate required

Cause: MODE. You tried to start a COM port without entering a baud rate. For example:

```
MODE COM1 PARITY=n DATA=8
```

Action: Re-enter the appropriate format specifying a baud rate.

Block mark required

Cause: E. You tried to create a box around a segment of text without marking it first.

Action: You must block mark the segment of text before you can create a box around it.

C

Cannot back up to 40/60MB tape in 80 /120MB tape drive

Cause: An 80MB drive can restore from, but not back up to, a 40MB tape.

Action: Try using an 80MB tape, formatting the tape, or using a blank tape.

Cannot change BUFSIZE

Cause: DOSKEY. The DOSKEY program is already installed.

Action: To change the BUFSIZE, reinstall the DOSKEY program using the /REINSTALL and the /BUFSIZE=size parameters.

If you need to conserve memory, you can restart your computer and install DOSKEY using just the /BUFSIZE=size parameters.

Note: Reinstalling with the /REINSTALL parameter does not remove the previous program from memory.

Cannot CHDIR to path - tree past this point not processed

Cause: CHKDSK. CHKDSK is checking the structure of the directory and is unable to go to the specified directory. All subdirectories underneath this directory cannot be verified.

Action: Run CHKDSK again with the /f switch to automatically correct this error.

Cannot CHDIR to root

Cause: CHKDSK. CHKDSK is checking the tree structure of the directory and is unable to return to the root directory. CHKDSK is unable to check the remaining subdirectories.

Action: Restart DOS. If this error persists, the disk is unusable and must be reformatted. See the documentation that came with your system.

Cannot CHKDSK a network drive

Cause: CHKDSK. This message indicates one of the following three problems:

- You cannot run CHKDSK against a remote server drive.
- You cannot run CHKDSK against a local drive that you are sharing on the network.
- You cannot run CHKDSK against a local drive on which you are logging network messages.

Action: If the disk is a remote server, you cannot run CHKDSK against it. CHKDSK must be run on the server itself. If the disk is being shared, you must issue two commands: NET PAUSE SRV and NET PAUSE PRT. After these commands are issued, you can run CHKDSK. If the network messages are being logged to your disk, issue the command NET PAUSE RCV (or whatever configuration you started your machine as); then run CHKDSK. If you wish to restart your network disk activities, issue the commands NET CONTINUE RCV (or whatever configuration you started your machine as), and NET CONTINUE PRT. These commands resume the possibility of any background disk activity.

Cannot CHKDSK a SUBSTed or ASSIGNed drive

Cause: CHKDSK. The SUBST and ASSIGN commands hide disk information which is necessary to CHKDSK.

Action: Remove the substitution. Then try the CHKDSK command again.

Cannot create a zero size partition.

Cause: FDISK. A partition must be a minimum of one cylinder.

Action: Specify a size for the partition of at least one cylinder.

Cannot create Extended DOS Partition without Primary DOS Partition on disk 1.

Cause: FDISK. You cannot have an extended DOS partition on disk 1 unless a primary DOS partition also exists.

Action: Use FDISK to create a primary DOS partition on disk 1. Leave enough room on the disk to allow space for the extended DOS partition.

Cannot create Logical DOS Drive without an Extended DOS Partition on the current drive.

Cause: FDISK. There is no extended DOS partition on this drive.

Action: Return to the FDISK Options menu and create an extended DOS partition.

Cannot delete Extended DOS Partition while logical drives exist.

Cause: FDISK. You cannot delete the extended DOS partition while there are logical DOS drives defined in that partition.

Action: Use FDISK to delete all the logical DOS drives in the extended DOS partition.

Cannot delete Primary DOS Partition on drive 1 when an Extended DOS Partition exists.

Cause: FDISK. An extended partition exists on drive 1. You cannot delete the primary DOS partition when there is an extended partition on the same drive.

Action: First delete the extended partition; then delete the primary DOS partition.

Cannot DISKCOMP to or from a network drive

Cause: DISKCOMP. You cannot use DISKCOMP to compare files on a network drive or on a drive that is on your system but is currently being shared on the network.

Action: Use COMP *.* instead of DISKCOMP. If the disk is being shared, you can PAUSE the server, do DISKCOMP, then CONTINUE the server.

Cannot DISKCOPY to or from a network drive

Cause: DISKCOPY. You cannot use DISKCOPY to copy files to or from a network drive or a drive that is on your system but is currently being shared on the network.

Action: Use XCOPY *.* instead of DISKCOPY. You can also use the COPY command to copy individual files instead of the complete disk. Make sure that you format your disk. If the disk is being shared, you can PAUSE the server, do DISKCOPY, then CONTINUE the server.

Cannot do binary reads from a device

Cause: COPY. You used the /B parameter with a device name while trying to copy from the device. The copy cannot be processed in binary mode because COPY must be able to detect the end-of-file from the device.

Action: Re-enter COPY and omit the /B parameter or use the /A parameter after the device name.

Cannot DRVLOCK a network drive

Cause: DRVLOCK. You tried to use DRVLOCK on a network drive. DRVLOCK supports only local drives.

Action: No action is required.

Cannot DRVLOCK a SUBSTed or ASSIGNed drive

Cause: DRVLOCK. You ran a SUBST or ASSIGN command before you used DRVLOCK. But, DRVLOCK does not support drives with these commands active.

Action: Remove the substitutions; then, try the DRVLOCK command again.

Cannot edit .BAK file--rename file

Cause: EDLIN. DOS considers files with the extension .BAK to be backup files. You usually do not edit .BAK files because more recent files exist.

Action: If it is necessary to edit the .BAK file, rename the file giving it an extension other than .BAK, copy the file, and give the copy a different file name extension.

Cannot EJECT a network drive

Cause: EJECT. You tried to use EJECT on a network drive. EJECT supports only local drives.

Action: No action is required.

Cannot EJECT a SUBSTed or ASSIGNed drive

Cause: EJECT. You ran a SUBST or ASSIGN command before you used EJECT. But EJECT does not support drives with these commands active.

Action: Remove the substitutions; then, try the EJECT command again.

Cannot EJECT media

Cause: EJECT. Drive ***d:*** is locked, the medium inside of the drive cannot be ejected, or due to some other failure.

Action: Use the DRVLOCK command with the /OFF parameter to unlock the drive.

Cannot FDISK with network loaded

Cause: FDISK. You tried to use FDISK while your system was connected to a network.

Action: Disconnect from the network before using FDISK.

Cannot find GRAPHICS profile

Cause: GRAPHICS. DOS cannot find the GRAPHICS.PRO file.

Action: Specify the path to the GRAPHICS.PRO file or DOS will look in the current directory.

Cannot find System Files

Cause: COMMANDS. Hidden files IBMBIO.COM, or IBMDOS.COM or both were not found on the current drive.

Action: Change the current drive to one that has the system files in the root and try again.

Cannot format an ASSIGNed or SUBSTed drive

Cause: FORMAT. You ran an ASSIGN or SUBST command before you used FORMAT.

Action: Process ASSIGN to restore the original drive letter assignment; then proceed with the FORMAT command.

Cannot FORMAT a network drive

Cause: FORMAT. You cannot use the FORMAT command to format a network drive or a drive on your system being shared on the network.

Action: If the drive is being shared, you can PAUSE the server, do FORMAT, then CONTINUE the server.

Cannot Loadhigh batch file

Cause: LOADHIGH. You tried to load a batch program (a file having a .BAT extension) into upper memory. You can load only executable or command programs (programs having either an .EXE or .COM extension) in upper memory.

Action: Do not use the LOADHIGH command with batch files. Remove all occurrences of the LOADHIGH command used with batch files from all directories and from your AUTOEXEC.BAT file.

Cannot lock drive *d:*

Cause: DRVLOCK. Due to a failure, DRVLOCK was not successful.

Action: Try DRVLOCK again.

Cannot lock socket *x:*

Cause: DRVLOCK. Due to a hardware failure, the lock function on the socket was not successful.

Action: Try the command again.

Cannot make directory entry

Cause: MKDIR. The disk or root directory is full. There can be no more than 512 files and directories in the root directory of your hard disk.

Action: Move or delete some files to free space, or use another disk.

Cannot move multiple files to a single file

Cause: MOVE. You tried to move multiple files to a single file.

Action: You can only move a single file to a single file. When moving more than one file, the destination must be a directory. If you want to combine multiple files into one file, see command "APPEND" on page 21.

Cannot move *x* to *y*: No such file or directory

Cause: MOVE. The target path is too long, the file or directory does not exist, or you are trying to move a directory to a subdirectory of itself. For example,

```
move c:\test c:\test\temp
```

Action: Specify a valid target.

Cannot open specified country information file

Cause: CHCP. The path to the COUNTRY.SYS file is not correctly specified.

Action: Check the country statement in your CONFIG.SYS file to make sure you specified the correct path to the COUNTRY.SYS file. Make any necessary changes and restart your computer.

Cannot perform a cyclic copy

Cause: XCOPY. The function requested causes the destination to be part of the source. The /S parameter was probably specified with XCOPY. The destination is a subdirectory, but the source is a directory above the destination.

Action: Evaluate the tree structure of the source and the destination and use a temporary file or disk to avoid the endless cycle.

Cannot read file allocation table

Cause: RECOVER. Recover cannot read the file allocation table (FAT) from the disk. The part of the disk where the FAT is held has been physically damaged and the FAT is irretrievable.

Action: If this message appears twice for FATs 1 and 2, format the disk to make it usable again. If FORMAT fails, the disk is probably unusable.

Cannot read system files from source

Cause: SYS. Hidden files IBMBIO.COM, or IBMDOS.COM or both were not found on the current drive or in the path.

Action: Change the current drive to one that has the system files in the root and try again.

Cannot RECOVER a network drive

Cause: RECOVER. You cannot use the RECOVER command to recover files from a network drive or from a drive on your system that is being shared on the network.

Action: If the drive is being shared, you can PAUSE the server, do RECOVER, then CONTINUE the server.

Cannot RECOVER an ASSIGNed or SUBSTed drive

Cause: RECOVER. You ran an ASSIGN, SUBST, or JOIN command before you used CHKDSK.

Action: Execute ASSIGN to restore the original drive letter assignment; then execute CHKDSK.

Cannot recover .. entry, Entry has a bad attribute (or link or size)

Cause: CHKDSK. The parent directory (..) is defective and cannot be recovered.

Action: If you specify the /F switch, CHKDSK tries to correct the error automatically.

Cannot recover .. entry, processing continued

Cause: CHKDSK. The working directory is defective and cannot be recovered.

Action: No action required. Processing continues.

Cannot setup expanded memory

Cause: FASTOPEN. The expanded memory card is not functioning properly in your system.

Action: If you cannot solve the problem, contact your authorized IBM representative.

Cannot specify default drive

Cause: SYS. Cannot specify the disk drive that you want to transfer the operating system files to.

Action: You must leave two unused entries in the root directory of the drive that you want to transfer the operating system files to, and you must leave enough space on the disk for SYS to install the system files.

Cannot unlock drive *d:*

Cause: DRVLOCK. Due to a failure, DRVLOCK was not successful.

Action: Try DRVLOCK again.

Cannot unlock socket *x:*

Cause: DRVLOCK. Due to a hardware failure, the unlock function on the socket was not successful.

Action: Try command again.

Cannot use FASTOPEN for drive *d:*

Cause: FASTOPEN. A JOINed, SUBSTed, or ASSIGNed disk drive or network drive was specified. FASTOPEN can only be used on hard disk drives.

Action: Remove the drive reassignment and specify the true drive letter, or specify only hard disk drive letters.

Cannot use PRINT – use NET PRINT

Cause: PRINT. You cannot use the PRINT command on a network server system.

Action: Use NET PRINT to print the files.

Cannot write file allocation table

Cause: RECOVER. Not determined.

Action: Use the CHKDSK command with the /F parameter to identify the problem. If you still get this message, you might have to reformat the disk. If you continue to get this message when checking a hard disk, contact your hardware vendor.

Cannot XCOPY from a reserved device

Cause: XCOPY. The source specified is a character device, such as a printer or an asynchronous communication port, or is NUL.

Action: Put the source data into a file and enter XCOPY again, using the file as the source.

Cannot XCOPY to a reserved device

Cause: XCOPY. The destination specified is a character device, such as a printer or an asynchronous communication port, or is NUL.

Action: Specify a file or block device other than NUL as the destination and enter XCOPY again.

Card Services is not installed

Cause: DRVLOCK. Card Services is not installed on your system.

Action: Check for PCMCIA support on your system. If your system has PCMCIA support, install Card Services.

Card Services already present

Cause: PCMCIA. You are attempting to install Card Services which is already installed.

Action: No action required.

Card Services not found!

Cause: PCMCIA. The Card Services driver is not installed.

Action: Install Card Services; then, try the command again.

CHKDSK /F cannot be done in a Windows/DosShell Command Prompt

Cause: CHKDSK. You tried to run the CHKDSK command with the /F parameter from the command prompt while running under the DOS Shell task swapper.

Action: Quit the DOS Shell before using the parameter with the CHKDSK command.

CHDIR . . failed, trying alternate method

Cause: CHKDSK. CHKDSK found an unrecoverable error while reading or writing to a subdirectory.

Action: Restart DOS and CHKDSK again.

CHKDSK not available on drive *X*

Cause: CHKDSK. You are trying to CHKDSK an alternate file system which cannot be found.

Action: Re-enter the statement using the correct parameters.

***x* code page driver cannot be initialized**

Cause: DISPLAY.SYS or PRINTER.SYS. You entered an invalid *type* in the CONFIG.SYS DEVICE statement.

Action: Re-enter the statement in the CONFIG.SYS file using the correct options to select from. See the information on DISPLAY.SYS and PRINTER.SYS in Chapter 5, "Working With Device Drivers" on page 245.

Code page not prepared

Cause: MODE. MODE failed during a SELECT operation for one of the following reasons:

- The indicated code page had never been defined to the device.
- The prepared code page does not have the correct font to support the current video mode.

Action: Use MODE PREPARE to pass the code page definition to the device; then reissue the MODE SELECT operation to the device. If the error still occurs, increase the number of subfonts (*m*) in the DEVICE=DISPLAY.SYS command in CONFIG.SYS, and restart DOS.

Code page *xxx* not prepared for all devices

Cause: CHCP. CHCP was unable to select the code page for one of the following reasons:

- A device was not prepared for the requested code page.
- A device I/O error occurred.
- The device does not support code page switching.

Action: Use the MODE command to prepare all devices for code page switching. Refer to "Loading a Code Page into Memory" in the *User's Guide*. Make sure the printer is online and not currently printing. Retry the CHCP command.

Code page *xxx* not prepared for system

Cause: CHCP. CHCP was unable to select the specified code page because no devices have been prepared with the MODE command.

Action: Make sure that all devices have been prepared for the code page using the MODE command. Refer to "Loading a Code Page into Memory" in the *User's Guide* and retry the command.

Code page operation not supported on this device

Cause: MODE. In an attempt to use MODE to PREPARE or SELECT a code page for a device, the specified "device" might have been a file name (not really a device), or it is not a device that supports code pages. The CONFIG.SYS file might not contain the DEVICE command to properly load the code page switching device driver. If the definition on the device is incorrect, the device cannot be initialized properly.

Action: Make sure that the device statement has no spelling errors. The specified "device" cannot be a file name. Change the CONFIG.SYS file, restart, and try the MODE command again.

Code page requested *xxx* is not valid for given keyboard code

Cause: KEYB. KEYB attempted to start a keyboard code page that is not valid for that keyboard. The *xxx* displays the requested code page.

Action: Using the MODE command, change the selected code page to one valid for the new keyboard, or specify a code page parameter in the KEYB command.

Code page specified has not been prepared

Cause: KEYB. You have a CON device driver loaded that supports code page switching. The code page specified has not been prepared for CON.

Action: Use the MODE command to prepare CON with the desired code page.

Code page specified is inconsistent with the selected code page

Cause: KEYB. The code page specified has been prepared for the CON device but is not the current CON code page. The specified code page is now the active keyboard code page. It is important to note that the selected CON code page is not changed. This means that your display and keyboard are operating in different code pages. The characters you type might be displayed incorrectly while your display and keyboard are operating in different code pages.

Action: Use the MODE command to change the selected code page for the display.

Command not known

Cause: E. You typed a command that is not recognized by the program, or you typed the name of the command incorrectly.

Action: Verify that you typed a valid command; then, try the operation again.

Compare error on side *n*, track *xx*

Cause: DISKCOMP. One or more locations on the indicated track and side contain different information when the disks are compared.

Action: This message is to inform you that there is a difference between disks. If you want an exact copy of a disk, use DISKCOPY.

Configuration too large for memory

Cause: IBMBIO. Occurs when the number specified for the /E parameter in the SHELL statement and either the FILES statement, the BUFFERS statement or both does not leave enough room for DOS to be loaded.

Action: Restart your computer with a different diskette. Then, in your CONFIG.SYS file, remove some devices, reduce the amount of memory used by devices, or reduce the number of files or buffers the system uses.

Now, start your computer again.

Content of destination lost before copy

Cause: COPY. The source file that you specified in the COPY command was overwritten before the copy process completed.

Action: No action required. Further copying proceeds normally.

Could not copy COMMAND.COM onto target disk

Cause: SYS. The SYS command was not able to copy the COMMAND.COM to the destination disk. The disk might be bad, the wrong type, or not formatted.

Action: Determine if the destination disk is usable. If it is not, use another disk.

Could not open output file *x*

Cause: QCONFIG. You tried to redirect output information without correctly specifying the **/o** option.

Action: Specify the name of the file you want to output information to using the **/o** switch.

Current drive is no longer valid

Cause: DOS. While attempting to get the current directory for the command prompt ($P), COMMAND.COM found that the disk could not be accessed. This can occur if your current drive is a network drive, or if the current drive is a diskette drive and there is no disk in it.

Action: If your current drive is a diskette drive, make sure that there is a diskette in the drive and the drive door is closed; otherwise, change your current drive to a valid drive.

Current keyboard does not support this code page
Cause: MODE. While attempting to carry out a MODE PREPARE or MODE SELECT code page function, the device has detected a discrepancy between the currently defined keyboard code and the specified code page.

Action: Adjust the KEYB specification to reference the desired code page definitions and reissue the MODE PREPARE code page function.

D

Data error
Cause: DOS. DOS was unable to read or write the data correctly (usually means that a disk has developed a defective spot).

Action: Refer to "Responses" on page 303 in this book.

Device error during Status
Device error during Prepare
Device error during Select
Device error during Refresh
Device error during write of font file to device
Cause: MODE. During the indicated operation of the MODE code page function, the device returned a device error. The device might not support code page functions, the device is not defined to contain enough code pages to meet the request, or the device detected certain types of invalidities within the font file contents. The CONFIG.SYS file might not contain the DEVICE command to properly load the device.

Action: Make sure that you specified the proper device name. Change the CONFIG.SYS file, restart, and try the MODE command again.

Device or code page missing from font file
Cause: MODE. After transmitting the specified font file to the device handler during a PREPARE operation, DOS responded with an error indicating that the font file does not contain a definition of the indicated code page for the specified device.

Action: Specify the MODE command indicating a different code page that is supported by the device. Check the description of the various font files to verify the device and code page combinations supported.

Note: The error causes the existing code pages to be undefined. All specified code pages must be prepared again.

Device *ddd* not prepared
Cause: MODE.
Where:
ddd = device name

In response to a MODE code page status request, the indicated device has not received any code page definitions through a MODE PREPARE command. This operation is required before carrying out the MODE SELECT operation.

Action: Information message. If desired, use the MODE PREPARE command to pass the code page definitions to the device.

Did not redirect file *x* successfully

Cause: QCONFIG. You tried to redirect output information without correctly specifying the **/o** option.

Action: Specified the name of the file you want to output information to using the **/o** switch. For syntax format information, see "QCONFIG" on page 193, or type **qconfig /?**

Directory already exists

Cause: MKDIR. You have tried to make a directory where one already exists by that name.

Action: Retry the MD command using another name.

Directory is joined

Cause: CHKDSK. You tried to use CHKDSK while the JOIN command was in effect. The CHKDSK command does not process joined directories.

Action: If you want to perform the CHKDSK process, use the JOIN command with the /D parameter at the command prompt.

Directory is totally empty, no . or ..,
tree past this point not processed

Cause: CHKDSK. A subdirectory was found that did not properly contain a . or .. entry. This usually happens when DOS is not given a chance to update the disk properly. During the updating process, the system might have shut down, or you might have reloaded the system before DOS finished the update.

Action: Use the RECOVER command to try to recover files on the damaged disk.

Directory not empty

Cause: JOIN. The directory you want to join to is not empty.

Action: Re-enter command with empty directory name or new directory.

Disk appears to contain data

Cause: The disk you are about to use appears to be a DOS-formatted disk that may contain data. Backup checks to see if the disk is formatted, but not whether it has any data.

Action: Choose OK to write on this disk, Cancel to abort this backup, or Retry after inserting another disk in place of this one.

Disk drive not known

Cause: COMMANDS. An invalid or nonexistent drive specification was used with the command or in one of its parameters, or the source and target drive are the same.

Action: Re-enter the command, using a valid drive specifier.

DISKCOPY cannot copy memory (CMCDD) devices

Cause: DISKCOPY. DISKCOPY in not supported on the memory device specified.

Action: Use COPY or XCOPY to transfer files to and from the memory device.

Disk error reading FAT *x*

Cause: CHKDSK. The file allocation table indicated is invalid. A power failure while a file is open can cause this.

Action: If this message appears twice for FATs 1 and 2, format the disk to make it usable again. If FORMAT fails, the disk is probably unusable.

Disk error reading (or writing) drive *x*:

Cause: DOS device error. DOS could not read the data from the disk properly. This is often due to a defective disk.

Action: Try typing R (for Retry) several times, or type A (for Abort) to end the program. Make a new copy of the disk.

Disk error writing FAT *x*

Cause: CHKDSK. A disk error was met while CHKDSK was attempting to update the file allocation table (FAT) on the specified drive. The variable, *x*, is 1 or 2, depending on which of the two copies of the file allocation table could not be written.

Action: If this message appears twice for FATs 1 and 2, format the disk to make it usable again. If FORMAT fails, discard the disk because it is probably unusable.

Disk full. Edits lost.

Cause: EDLIN. An End Edit command ended abnormally because the disk was full (not enough free space to save the complete file). Any editing done to the file is lost.

Action: Get a new disk, copy the file onto the new disk, and start editing again.

Disk unsuitable for system disk

Cause: FORMAT. A defective track was detected where the DOS files were to reside.

Action: The disk can be used only for data. Use another disk if you wish to make a system disk.

Divide overflow

Cause: DOS. A program tried to divide a number by 0, or a logic error caused an internal malfunction. The program ends and returns you to DOS.

Action: Correct the programming error and continue. If this is a purchased program, take it back to your authorized representative.

Does not exist

Cause: CHKDSK. If you specify the /V parameter, CHKDSK attempts to display every action during its processing. This message indicates that the . or .. subdirectory entry previously displayed could not be found. This means that the entire directory could be invalid. It might also mean that only a single byte on the disk is invalid.

Action: Run CHKDSK again with the /f switch. CHKDSK will attempt to correct the problem and return the directory to its normal state.

Do not specify filename(s)

Cause: DISKCOMP or DISKCOPY. At least one file name was specified with the DISKCOMP or DISKCOPY command.

Action: Specify only the drive letters of the disks, not the file name.

Drive already SUBSTed

Cause: SUBST. You tried to substitute a drive that was already substituted. A drive can only be substituted once.

Action: If you want to change the current substitution, use the SUBST command with the /D parameter at the command prompt, and use the SUBST command again.

Drive *d:* already deleted.

Cause: FDISK. The drive you selected to delete has already been deleted.

Action: Type in a different drive letter or press Esc to return to the FDISK Options menu.

Drive *d:* does not support DRVLOCK

Cause: DRVLOCK. The specified drive does not support the lock or unlock functions of DRVLOCK.

Action: Refer to the documentation that came with the drive to verify if it supports the DRVLOCK functions.

Drive *d:* does not support EJECT

Cause: EJECT. The specified drive does not support EJECT.

Action: Make sure the drive you specify supports the EJECT function.

Drive *d:* is locked

Cause: EJECT. You tried to used EJECT on a drive that is locked.

Action: Use the DRVLOCK command with the /OFF parameter to unlock the drive. Then, try the EJECT command again.

Drive letters have been changed or deleted

Cause: FDISK. One or more logical DOS drives have been deleted from the extended DOS partition. The drive letter assignments for any remaining logical drives might have changed, because DOS assigns drive letters to logical DOS drives based on the physical location of these drives in the extended DOS partition.

Action: Note the new drive letter assignments on the FDISK option for the new drive letter assignments.

Drive or device not ready

Cause: COMMANDS. An error occurred while trying to read or write to the indicated drive. Common causes of this error are as follows:

- The diskette drive door is not closed.
- The diskette is not properly formatted.
- The printer is not online.
- The printer does not have enough paper or is busy.

Action: Correct the error and try again. If your default drive is a diskette drive, try typing F to return to the command prompt.

Drive types or diskette types not compatible

Cause: DISKCOMP or DISKCOPY. You must have the same size and type of diskettes to run these commands. For example you cannot copy from a single-sided diskettes to a double-sided diskette, or compare a high-density diskette with a low-density diskette.

Action: You should use FC if you want to compare the files on the diskettes. If you want to copy the diskette, try COPY or XCOPY. Refer to the DISKCOMP or DISKCOPY commands for the allowable combinations.

Duplicate file name or file not found

Cause: RENAME. You tried to rename a file to a file name that already exists on the disk, or the file to be renamed could not be found on the specified (or default) drive. RENAME is warning you that you are using the same name for two files, or it cannot find the file you are trying to rename.

Action: Make sure that you typed the file name correctly. Re-enter the RENAME command.

Duplicate parameters not allowed

Cause: GRAPHICS. You specified a parameter twice.

Action: Re-enter the command using the correct format.

E

EMM386: Unable to start Enhanced Mode Windows due to invalid path specification for EMM386

Cause: EMM386. The EMM386 path in the CONFIG.SYS is invalid.

Action: Edit your CONFIG.SYS file and type the correct drive and path name on the EMM386 line.

Entry error

Cause: EDLIN. EDLIN has detected a syntax error.

Action: Correct the syntax error on the last command.

Entry has a bad attribute (or size or link)

Cause: CHKDSK. This message might begin with one or two periods, indicating which entry in the subdirectory was in error. One period shows that the current directory is in error. Two periods mean that the parent directory is in error. If you did not enter the /F parameter, no corrective action is taken.

Action: Type:

```
CHKDSK /F
```

CHKDSK tries to correct the error.

Entry has a bad link
Entry has a bad size

Cause: CHKDSK. The "." (current directory) or ".." (parent directory) entry contains incorrect information.

Action: Use the CHKDSK command with the /F parameter to correct the error.

ERROR: Bad command line, no installation

Cause: PCMCIA. The specified command line parameter is not correct.

Action: Specify the correct command line parameter; then, try the command again. For more information, see command "PCMCS" on page 178.

ERROR: Card Services not found

Cause: PCMCIA. The Card Services driver is not installed.

Action: Install the Card Services driver; then, try the command again. For installation information, see the *User's Guide.*

ERROR: Card Services not installed

Cause: PCMCIA. The Card Services driver is not installed.

Action: Install the Card Services driver; then, try the command again.

ERROR: Could not register as client

Cause: PCMCIA. An attempt to register PCMSCD was not successful. Possible causes might be that the default limit of ten client drivers has been reached, or a number smaller than what is needed was specified during the installation of Card Services.

Action: Use the **/client** switch with the PCMCS command to increase the number of clients that can be registered; then, try the command again.

ERROR: PCMSCD is already loaded

Cause: PCMCIA. You are attempting to install Card Services which is already loaded.

Action: No action required. PCMCSD is already installed.

Error CPU running too slow for overlap mode

Cause: This system cannot handle the high data rates necessary for high-speed DMA mode.

Action: Therefore, Backup automatically drops to the /NO option mode with no overlapped DMA.

Error decompressing file

Cause: An error was found during the decompression of this file while restoring. The file is probably corrupted.

Action: This should not happen if you backed up with verify on unless the disk was damaged sometime after the backup.

Error during read of font file

Cause: MODE. While attempting to carry out a MODE PREPARE code page function, the record from the specified code page font file had an unrecoverable I/O error. The code page definition sent to the device is incomplete.

Action: Try to determine the readability problem and restore the file from a master copy, or direct MODE to an alternate device to access a copy of the file.

Error General Hardware Failure xx

Cause: You may see this message followed by a specific number. These messages generally indicate faulty hardware, wrong configuration, or the need to use the /NO parameter.

Action: Try the following:

- Check your setup to be sure you have selected the correct drive type and media types.
- Check your drive to be sure the disk is inserted properly and the drive latch is secure.
- Determine if you have any memory-resident programs that are interfering with the operation of Backup.

- Try running Backup with the /NO parameter. This turns off the high speed DMA overlap, which is the same as setting the backup speed to medium.

Error in CONFIG.SYS line *x*

Cause: IBMBIO. An error occurred during startup when DOS reads the CONFIG.SYS file and interprets the commands within the file. The *x* variable shows you where the error occurred.

This message might also be shown when a device driver attempts to install itself but is not needed by the system.

Action: Edit the CONFIG.SYS file and re-enter the correct format of the command.

Error in COUNTRY command

Cause: DOS. The code page was missing from the COUNTRY command and the default was not requested or the country information file:

- Is not in correct format.
- Contains invalid information.

Action: Make sure that the country information is correct, or correct the COUNTRY command in the CONFIG.SYS file and restart.

Error in EXE file

Cause: DOS. An error was detected in the relocation information placed in the file by the LINK program. This might be caused by a change to the file.

Action:

- If you are using a purchased program, rerun the program, using your backup copy.
- If you are using a program you wrote yourself, go through the LINK procedure again.

Error in margin settings

Cause: E. Possible causes are as follows:

- You tried to set the margin past the maximum margin column.
- You tried to set a margin in the zero or negative column.
- You tried to set the left margin in a column greater than the right margin.

Action: Refer to the *PC DOS User's Guide* for more information about the appropriate margin settings.

Error loading operating system

Cause: DOS. A disk error occurred while attempting to load your operating system from the hard disk.

Action: Restart the system. If the error persists after several tries, restart the system (you should start DOS from your DOS diskette), and use the SYS command to transfer a new copy of DOS to your hard disk.

Error not a CP Backup disk

Cause: The disk you are attempting to restore from was created by an earlier version of Backup, or you may have the wrong backup speed selected.

Action: Low-speed backups can only be restored at low speed. High- or medium-speed backups may be restored at either high or medium speed. Also, check the media settings to be sure they are correct.

Error occurred in environment variable

Cause: DIR. The command line error which appeared just before this message occurred because the value of the DIRCMD environment variable is not proper DIR command syntax.

Action: Check the description of the DIR command or type DIR /? to display the proper command syntax. Use the SET command to view, change, or clear the value of the DIRCMD environment variable.

Error on device *d*

Cause: COMMANDS. An error occurred while trying to read or write to the indicated drive. Common causes of this error are as follows:

- The diskette drive door is not closed.
- The diskette is not properly formatted.
- The printer is not online.
- The printer does not have enough paper or is busy.

Action: Correct the error and try again. If your default drive is a diskette drive, try typing F to return to the command prompt.

Error on drive *d:*

Cause: COMMANDS. An error occurred while trying to read or write to the indicated drive. Common causes of this error are as follows:

- The diskette drive door is not closed.
- The diskette is not properly formatted.
- The printer is not online.
- The printer does not have enough paper or is busy.

Action: Correct the error and try again. If your default drive is a diskette drive, try typing F to return to the command prompt.

Error on hard disk *d:*

Cause: FDISK. The FDISK program was unable to read the startup record of the current hard disk after five attempts.

Action: Try the FDISK program again. If, after several attempts, you get the same error, consult the documentation that came with your system.

Error opening file

Cause: E. The file you are trying to open might not exist.

Action: Verify that you used the correct file name and the correct path that leads to the file that you want to open.

Error reading backup directory
Backup directory not found

Cause: These errors indicate that Backup either could not find a history file on the disk inserted as the last disk or that the history file was damaged.

Action: If this was not the last disk of the backup set, choose Retry and insert the correct disk. If you want to proceed with a restore without a good last disk, choose Rebuild. You will be prompted for the first disk of the backup set. The history file is then reconstructed from the entire set of backup disks.

Error reading directory

Cause: FORMAT. There are bad sectors within the directory or file allocation table (FAT) structure. An error occurred when accessing the directory.

Action: If you are trying to format your hard disk, try formatting the disk to make it usable again.

Error reading hard disk

Cause: FDISK. The FDISK program was unable to read the startup record of the current hard disk after five attempts.

Action: Try the FDISK program again. If, after several attempts, you get the same error, consult the documentation that came with your system.

Error reading GRAPHICS profile

Cause: GRAPHICS. DOS cannot read the GRAPHICS profile.

Action: Make sure that your disk drive door is closed.

Error reading partition table

Cause: FORMAT. An error (probably hardware) occurred while reading the partition table.

Action: Run FDISK on the drive and try again.

Error reading the file

Cause: UNFORMAT. While attempting to restore the partitions of a hard disk, the UNFORMAT command was unable to locate, or read the PARTNSAV.FIL file on the diskette.

Action: Make sure that the file exists on the diskette and try the UNFORMAT command again with the /PARTN parameter.

Error Track 0 bad or check your drive and media settings

Cause: This error may indicate:

- a damaged disk - try another disk.
- that you are using the wrong backup speed. Try a different setting.
- a conflict with a TSR.

Action: Check your drive and media settings and be sure you are using the correct media.

Note: Try using Media Format _ DOS Format if you are experiencing trouble on an Amstrad computer.

Error while performing Tape Tools Command

Cause: This is a general message that appears if something went wrong when Backup attempted to carry out the command you selected.

Action: Check the tape and the drive to ensure it is properly connected and seated in the drive. Try the command again.

Error writing directory

Cause: FORMAT. There are bad sectors within the directory or file allocation table (FAT) structure. This is a write problem.

Action: Try formatting the disk to make it usable again.

Error writing FAT

Cause: FORMAT. A disk error occurred while FORMAT was attempting to update the file allocation table (FAT) on the specified drive.

Action: If this message appears twice for FATs 1 and 2, format the disk to make it usable again.

Error writing file

Cause: E. The file might be read-only or there is not enough available free storage space left on the disk.

Action: Do one of the following:

- Specify a file that can be written to.
- Make storage space available by erasing unneeded files.
- Use the ATTRIB command to change the attributes of the file you want to write to.

For information about changing file attributes, see "ATTRIB" on page 26.

Error writing hard disk

Cause: FDISK. The FDISK program was unable to write the startup record of the current hard disk after five attempts.

Action: Try the FDISK program again. If, after several attempts, you get the same error, consult the documentation that came with your system.

Error writing partition table

Cause: FORMAT. An error (probably hardware) occurred while writing the partition table.

Action: Run FDISK on the drive and try again.

Error writing to device

Cause: COMMANDS. DOS met an I/O error when writing output to a device. The device is unable to handle the number of bytes requested.

Action: Change the amount of data in the file and retry the command.

Error writing track 0, unable to recover (Check Setup)

Cause: This message indicates that Backup was unable to successfully write to and verify the first sector of the current disk.

Action: Either the disk is defective or perhaps your setup is not correct for the type of disk drive installed. (For example if you have a 1.2M drive and it is set up as a 360K drive.)

ERROR - Input file *x* cannot be expanded onto itself

Cause: EXPAND. You tried to expand a file onto itself. You can only expand a file to a different file name.

Action: Try the command again, using another file name to expand the file to.

ERROR - Input parameters are not valid.

Cause: EXPAND. The specified parameters are not valid.

Action: Make sure that the syntax is valid for the EXPAND command; then, try the command again. For information about the EXPAND command syntax, see command “EXPAND” on page 102, or type **expand /?** at the DOS command line prompt.

Errors found, F parameter not specified
Corrections will not be written to disk

Cause: CHKDSK. Information message. An error was found and you have not used the /F parameter. CHKDSK carries out its analysis as though it were going to correct any errors detected, so that you can see the results of its analysis, but it does not write the corrections to the disk.

Action: Use the **/f** parameter if you want CHKDSK to try to correct any errors detected.

Errors on list device indicate that it may be off-line. Please check it.

Cause: PRINT. The printer was not recognized as being online.

Action: Ensure that the printer is plugged in, switched on, and that it is ready to print.

Error reading file

Cause: E. The editor does not recognize the file format used.

Action: You can only edit ASCII text files. You cannot edit files created with word processing programs.

EXEC failure

Cause: DEBUG. DOS either found an error when reading a command, or the FILES command in the CONFIG.SYS file is set too low.

Action: Increase the value of the FILES command in the CONFIG.SYS file, and restart DOS.

Expanded memory not available

Cause: COMMANDS. Information message. You have specified the **/x** or **/a** switch but you do not have an expanded memory card installed in your system.

Action: Remove the **/x** or **/a** switch from the command.

Extended DOS Partition already exists.

Cause: FDISK. An extended partition for DOS has already been set up.

Action: Return to the FDISK Options menu and select a different option.

Extended error #

Cause: DOS. An error occurred on the command line.

Action: The “#” position in this message represents a number that corresponds to a particular error that caused this message to be displayed.

The specific causes and actions for these errors are listed in this book in alphabetic order. Use the following list to find the message that corresponds to the number displayed on your screen:

1. - Invalid Function
2. - File not found
3. - Path not found

4. - Too many open files
5. - Access denied
6. - Invalid handle
7. - Memory control blocks destroyed
8. - Insufficient memory
9. - Invalid memory block address
10. - Invalid Environment
11. - Invalid format
12. - Invalid function parameter
13. - Invalid data
15. - Invalid drive specification
16. - Attempt to remove current directory
17. - Not same device
18. - No more files
19. - Write protect error
20. - Invalid unit
21. - Not ready
22. - Invalid device request
23. - Data error
24. - Invalid device request parameters
25. - Seek error
26. - Invalid media type
27. - Sector not found
28. - Printer out of paper error
29. - Write fault error
30. - Read fault error
31. - General failure
32. - Sharing violation
33. - Lock violation
34. - Invalid disk change
35. - FCB unavailable
36. - System resource exhausted
38. - Out of input
39. - Insufficient disk space
80. - File exists
82. - Cannot make directory entry
83. - Fail on INT 24
84. - Too many redirections
85. - Duplicate redirection
86. - Invalid password
87. - Invalid parameter
88. - Network data fault
90. - Required system component not installed

F

Fail on INT 24

Cause: DOS. The program stopped because it received a failure response to an interrupt message. This could occur when you choose **Fail** in response to the `Abort, Retry, or Fail` message.

Action: Correct the problem that caused the error, and try the command again.

Failure to access code page font file

Cause: MODE. During a PREPARE, the attempt to access the indicate code page font file failed.

Action: Make sure that the font file name is spelled correctly. Verify the presence of the specified font file. Reissue the MODE command with the proper specification of the font file name.

Failure to access device: *ddd*

Cause: MODE. During a code page operation, the opening of the specified DEVICE failed.
Where:
ddd = device name

Action: Make sure that the DEVICE name is spelled correctly and reissue the MODE command with the proper spelling. If that fails to correct the situation, make sure that the specified DEVICE was loaded by the CONFIG.SYS DEVICE command, or that the DEVICE is a standard device always present. If the CONFIG.SYS is incorrectly specified, edit that file to the proper specification and restart before retrying the MODE command.

FASTOPEN already installed

Cause: FASTOPEN. The FASTOPEN program is already installed and active on your system.

Action: If you want to specify different parameters, restart your computer and start the FASTOPEN program using the parameters you want.

FASTOPEN cannot be installed under DosShell

Cause: FASTOPEN. You attempted to install the FASTOPEN program from a command prompt while running under the DOS Shell task swapper option. The FASTOPEN program can be installed only when the task swapper is not enabled. **Warning:** Because FASTOPEN can lock your machine, it is not advisable to install FASTOPEN while you are running the IBM DOS Shell.

Action: To use FASTOPEN with the IBM DOS Shell, quit the DOS Shell, install the FASTOPEN program; then, start the DOS Shell again.

FASTOPEN EMS entry count exceeded. Use fewer entries

Cause: FASTOPEN. You have tried to set up FASTOPEN in EMS, but there was not enough space in EMS.

Action: If you want to set up FASTOPEN in EMS, enable FASTOPEN with a reduced size of file/directory entries and extent entries.

Fatal Error: Cannot allocate Memory for DOS

Cause: DOS. The conventional memory size in your system is too small. You need at least 128KB of conventional memory to start DOS.

Action: Install the required additional memory in your system.

Fatal Error; RTLink CACHE -Expanded memory handling

Cause: This message indicates a memory problem.

Action: Set the following environment variables before running Backup:

```
set rtvmexp=0

set rtvmext=0
```

FCB unavailable

Cause: COMMANDS. An application tried to access a file using a File Control Block (FCB) over a network or while the SHARE program was loaded; however, the FCB was no longer available.

Action: End the process. If this problem occurs frequently, increase the value in the FCB statement in your CONFIG.SYS file and restart your computer.

File allocation table bad, drive *d*

Cause: DOS and COMMANDS. You tried to read absolute sectors on a network drive, or the disk has been destroyed.

Action: If possible, use the Loader Write filespec option of the DEBUG command.

If this error persists, the disk is unusable and should be formatted again.

File cannot be converted

Cause: EXE2BIN. The input file is not in the correct format.

Action: Check the file to verify format.

File cannot be copied onto itself

Cause: COPY, XCOPY, or REPLACE. You tried to copy a file and place the copy (with the same name as the original) in the same directory and on the same disk as the original file.

Action: Change the name given to the copy, or put it in a different directory, or put it on another disk.

File creation error

Cause: DOS and COMMANDS. An unsuccessful attempt was made to replace a file that was already there or to add a new file name to the directory.

Action: Proceed with one of the following:

- If the file was already there, make sure that the file is marked “read only” and cannot be replaced.
- If you are not able to add a new file name to the directory, the maximum number of directory entries to be copied to the root directory might have been reached. Try removing a file, subdirectory, or the volume label from the root directory.
- Run CHKDSK to determine if some other condition caused the error.

File exists

Cause: COMMANDS. You tried to add a file to your system, but the file name you specified already exists.

Action: Select another directory or file name.

File is READ-ONLY

Cause: EDLIN. The specified file is read-only.

Action: Use the ATTRIB command to change the attribute to read-write.

File missing

Cause: Once a backup has started, the directory and file structure for the specified drive is in memory. If you delete any files while the backup is progressing, this message appears because Backup does not find a file where it "remembers" it should be.

Action: If you want the file included in the backup, restore it at a location you want and start the back up process again. Or, you can continue the back up process and not have the file included.

File name must be specified

Cause: EDLIN. You did not specify the name of the file you want to create or edit.

Action: Type the command and the file name at the command line.

File not found

Cause: E. You did not give the file a name before attempting to save it.

Action: Use the **F7** key to give the file a name then save it.

File not found - *filename*

Cause: DOS and COMMANDS. A file name in a command or command parameter does not exist in the directory on the specified (or default) drive.

Action: Retry the command, using the correct file name.

File not in PRINT queue

Cause: PRINT. You tried to remove a file that does not exist from the print queue.

Action: Type the correct name of the file you want removed from the print queue.

First Allocation unit is invalid, entry truncated

Cause: CHKDSK. The file whose name precedes this message contains an invalid pointer to the data area. If you specify the /F parameter, the file is truncated to a zero-length file.

Action: No action is required.

FIRST diskette bad or incompatible

Cause: DISKCOMP. The errors occurred while reading the diskette. The error might have been caused by bad sectors or the diskette in the source drive might not be compatible with the source drive type (a high-capacity diskette in a 320/360KB diskette drive, for example).

Action: Check your diskette by using the CHKDSK command.

Fixups needed - base segment hex:

Cause: EXE2BIN. The source file contained information indicating that a load segment is required for the file.

Action: You must specify the absolute segment address where the finished module is to be located.

Font file contents invalid

Cause: MODE. After transmitting the specified font file to the device driver during a PREPARE operation, DOS responded with an error indicating that the font file is not in the proper format. The file indicated might not be a font file, or the font file has been damaged, altered, or truncated.

Action: Make sure that you specified the correct name of the font file, and repeat the MODE command with the proper spelling of the font file. Compare the font file with the master copy to see if it has been altered, and replace it with a new copy from the DOS diskette.

Note: The error causes the existing code pages to be undefined. All specified code pages need to be prepared again. Refer to the MODE command.

FOR cannot be nested

Cause: Batch file. More than one FOR subcommand was found on one command line in the batch file.

Action: Use only one FOR subcommand per command line; then retry the command.

FORMAT not supported on drive *d:*

Cause: FORMAT. The disk device driver does not support or incorrectly handles generic IOCTL calls (function call 44 AL=0DH) for Get device parameters, and either Set device parameters or Format/Verify/Read/Write Track, or both. Virtual drives are set up pre-formatted and cannot be reformatted.

Action: If you have replaced the default disk device driver at setup time, remove that DEVICE command from your CONFIG.SYS file and try again. If you are using only the default disk device drivers, reinstall DOS on the disk you restart from and try again.

Function not supported on this computer - *x*

Cause: MODE. Your system does not have the appropriate adapter or device for the requested function.

Action: No action is required. Make sure that the EGA switch settings are in the enhanced color mode.

G

General failure

Cause: COMMANDS. A disk or device error has occurred, common causes of this error are:

- Diskette type and drive type do not match (for example, a 1.2MB high-capacity diskette in a 320/360KB drive).
- The diskette is not completely inserted in the drive or the drive door is open, or both.
- The disk is not properly formatted.

Action: Refer to “Responses” on page 303 of this book. Select the response Retry first, then select Abort if this problem requires further investigation by a programmer.

If you are using a purchased program, contact the representative you purchased it from.

GETCSDLEVEL - Could not open CSD file

Cause: QCONFIG. An attempt to open a CSD file was unsuccessful.

Action: No action required.

GETCSDLEVEL - Did not read anything from CSD file

Cause: QCONFIG. An attempt was made to read an empty CSD file.

Action: No action required.

GETVIDEO - Could not determine Video - Reg

Cause: QCONFIG. The video mode could not be determined.

Action: No action required.

H

Has invalid allocation unit, file truncated

Cause: CHKDSK. The file name preceding this message means that the file contains an invalid pointer to the data area.

Action: Use the /F parameter to truncate the file at the last valid data block. No corrective action occurs if CHKDSK is processed without the /F parameter.

Help not available for this command

Cause: HELP. The specified command is not spelled correctly or there is no help available for the command.

Action: Type the correct command name.

HMA not available: loading DOS low

Cause: DOS. Your CONFIG.SYS contains the line DOS=HIGH, but DOS cannot load itself into high memory because none is available.

Action: This could occur if your CONFIG.SYS did not contain the line DEVICE=HIMEM.SYS.

I

Illegal device name

Cause: MODE. The specified printer must be LPT1, LPT2 or LPT3. The specified Asynchronous Communications Adapter must exist and must be COM1, COM2, COM3, or COM4.

Action: Use the correct device name and retry the command.

Incorrect APPEND version

Cause: APPEND. You are using a version of APPEND that does not match the version you initially loaded. You might be using the version that comes with the IBM Local Area Network Program.

Action: Determine why you are accessing the wrong version (often the result of your PATH setting) and change your setup to find the DOS version of APPEND.

Incorrect command-line parameter(s) detected

Cause: PCMCIA. The specified command line parameter is not correct.

Action: Specify the correct command line parameter; then, try the command again. For information about the command, check in the "Command Reference" part of this book.

Incorrect DOS version

Cause: DOS. The command you attempted to use is not intended to be used with the version of DOS now on your system.

Action: If you set up with DOS, make sure that the DOS commands you use are from the DOS diskettes. Network users accessing DOS commands across the network should edit the PATH statement in the NETPATH.BAT file, replacing the "*d*:\APPS\DOS" reference with a local reference to DOS; for example, A:\ or C:\DOS.

Incorrect number of parameters

Cause: COMMANDS. You entered a different number of parameters than is valid for the command.

Action: Correct the command parameters and try again.

Incorrect parameter

Cause: FASTOPEN. A parameter was specified that was not recognized.

Action: Specify the correct command line parameter; then, try the command again. For information about the command, check in the "Command Reference" part of this book or type at the DOS command line prompt the name of the command followed by /**?**.

Infinite retry not supported on network printer

Cause: MODE. Infinite retry was requested by specifying P or RETRY=. Printer errors cannot be sensed through the network interface.

Action: Do not specify P or RETRY= or use a non-redirected printer.

Insufficient disk space

Cause: DOS and COMMANDS. The disk does not contain enough free space to contain the file being written.

Action: Refer to "Responses" on page 303 of this book. If you suspect this condition is invalid, run CHKDSK to determine the status of the disk. Otherwise, use another disk and retry the command.

Insufficient disk space for temporary file

Cause: Backup needs to create temporary files while processing.

Action: The drive must not be write-protected and must have space available (approximately 80K per 20M of disk being backed up or 120K if using a SCSI device).

Note: If you are using a RAM drive for temporary files, be sure it is big enough to contain them. The temporary files are written to the directory specified by the environment variables: TEMP=, TMP=, or directly to the root directory of the first local nonremovable drive.

Insufficient memory

Cause: COMMANDS. The amount of available memory is too small to allow these commands to function.

Action: You need to increase the amount of available memory in your system. You can do this in several ways including:

- Changing the BUFFERS parameter in the CONFIG.SYS file to a smaller value.
- Removing terminate and stay resident programs.

Restart the system and try the command again. If the message still appears, your system does not have enough memory to process the command.

Insufficient memory for COUNTRY.SYS file

Cause: IBMBIO. There is not enough memory for processing the COUNTRY.SYS file.

Action: You need to increase the amount of available memory in your system. You can do this in several ways including:

- Change the BUFFERS parameter in the CONFIG.SYS file to a smaller value.
- Remove device driver programs.

Restart the system and try the command again. If the message still appears, your system does not have enough memory to process the command.

Insufficient memory to load system files

Cause: COMMANDS. There was not enough memory to keep all of the system files in random access memory (RAM).

Action: Free memory by closing files and programs you no longer need. If you still get this message, remove memory-resident programs or unnecessary device drivers from your CONFIG.SYS file, and restart your computer.

Insufficient memory to store macro. Use the DOSKEY command with the /BUFSIZE switch to increase available memory.

Cause: DOSKEY. There is not enough memory to store the information you typed at the command prompt.

Action: Use the DOSKEY command with the /MACROS parameter to see a list of the macros currently resident in memory. To free some memory for the new macro, delete the macros you no longer need.

If you still get this message, install the DOSKEY program again.

Note: Any macros not saved in a batch file will be erased.

To reinstall the DOSKEY program, do one of the following:

- Restart your computer and specify the DOSKEY command with a larger /BUFSIZE parameter. Or,
- Without restarting your computer, use the DOSKEY command with the /REINSTALL and the /BUFSIZE parameters.

Insufficient room in root directory
Move files from root directory and repeat CHKDSK.

Cause: CHKDSK. You instructed CHKDSK to create files from the “lost” data blocks it has found, but the root directory is full, and all the lost chains could not be recovered into files.

Action:

1. Copy some of the recovered files to another disk for further examination.
2. Delete the recovered files from the disk you are checking.
3. Run CHKDSK again to recover the remainder of the lost data.

Intermediate file error during pipe

Cause: DOS. DOS is unable to create one or both of its intermediate files because the root directory default drive was full. DOS is unable to find the piping files, or the disk does not have enough space to hold the data being piped.

Action: Erase some files from the root directory of the default drive, and reissue the command that failed. If you get the same message, one of the programs in the command line has erased one or both piping files. Correct the program and reissue the command line.

Internal error

Cause: FDISK or SHARE. This message indicates an undetermined error has occurred.

Action: Contact your service representative.

Invalid baud rate specified

Cause: MODE. You specified an invalid baud rate.

Action: Specify the baud rate as 110, 150, 300, 600, 1200, 2400, 4800, 9600, or 19200 (you need specify only the first two characters of the number).

Invalid characters in volume label

Cause: FORMAT or LABEL. The volume label contains one or more invalid characters, or the name contained a period.

Action: Specify a volume label with up to 11 characters, using only letters and numbers.

Invalid code page

Cause: DOS and COMMANDS. The code page specified on the command line might not be used with the requested language.

Action: Specify the correct code page and retry the operation.

Invalid code page specified

Cause: KEYB. You specified the wrong code page setting with the MODE command /SELECT parameter, or the code page setting in your CONFIG.SYS file is not correct for your keyboard layout.

Action: Specify the correct code page value for your keyboard layout using the MODE command with the /SELECT parameter, or edit your CONFIG.SYS file to specify the correct value; then restart your computer.

Invalid combination of parameters

Cause: COMMANDS. One or more of the parameters entered have been placed in the wrong order.

Action: Review the syntax for the command and enter the command again.

Invalid command line!

Cause: PCMCIA. The specified command line parameter is not correct.

Action: Specify the correct command line parameter; then, try the command again. For information about the command, check in the "Command Reference" part of this book.

Invalid country code or code page

Cause: IBMBIO. The three-digit country code or code page specified in the COUNTRY command in the CONFIG.SYS file is invalid or the requested code page is not available for the specified country code.

Action: See the command "COUNTRY" on page 51 in this book for the correct country code and code page.

Invalid current directory

Cause: CHKDSK. CHKDSK attempted to read the current directory and found an unrecoverable error on the disk.

Action: You might be able to recover some of the files on this disk by copying them with the copy command.

Invalid date

Cause: DOS and COMMANDS. The date format used was not correct for the country code setting in your CONFIG.SYS file.

Action: Type the date using the correct format for the country code setting specified in your CONFIG.SYS file, as indicated by the screen prompt.

Invalid device

Cause: CTTY. DOS does not recognize the device name specified.

Action: Retry the command, using a valid device name.

Invalid device parameters from device driver

Cause: FORMAT. The number of hidden sectors (relative sectors) is not an exact multiple of the number of sectors per track. The DOS partition does not start on a track boundary.

Action: Load FDISK and set up a new DOS partition on the hard disk. Retry FORMAT.

Invalid device request

Cause: COMMANDS. A device driver has issued an invalid command to <*device*>.

Action: Refer to "Responses" on page 303 in this book.

- Review your device interface specification and DOS driver implementation to make sure that everything you are trying to do is supported.
- Check your program to see if you have a coding problem that needs debugging.

Invalid directory

Cause: DOS and COMMANDS. One of the directories in the specified path does not exist.

Action: Retry the command, using a valid directory.

Invalid disk change

Cause: COMMANDS. You changed a disk while an operation was in progress.

Action: Insert the original disk in the drive and try the command again.

Invalid drive in search path

Cause: DOS. An invalid drive specifier was found in one of the paths specified in the PATH command. This message appears when DOS attempts to find a command or batch file rather than at the time the incorrect PATH command was issued.

Action:

1. Enter PATH. This displays the paths previously defined.
2. Find the invalid specifier.
3. Re-enter the PATH command with the valid drive specifier and the desired paths.

Invalid drive or file name

Cause: DOS or COMMANDS. The drive or file name specified is invalid.

Action: Enter a valid drive name or file name.

Invalid drive specification

Cause: COMMANDS. An invalid or nonexistent drive specification was used with the command or in one of its parameters, or the source and target drive are the same.

Action: Re-enter the command, using a valid drive specifier.

Invalid drive specification
Specified drive does not exist,
or is non-removable

Cause: DOS or COMMANDS. An invalid drive specification was just entered in a command or in one of its parameters.

Action: Re-enter the command, using a valid drive specifier.

Invalid entry

Cause: FDISK. You made an invalid entry.

Action: Type the correct value.

Invalid entry, please enter *x*.

Cause: FDISK. The value entered was not within acceptable range.

Action: Re-enter your choice.

Invalid entry, please press Enter.

Cause: FDISK. The percent sign you entered is the last character. You cannot enter additional characters after you have entered the percent sign.

Action: Press Enter, or press Backspace to change your entry and press Enter.

Invalid Environment

Cause: COMMANDS. This message is issued when:

- The specified environment size contains non-numeric characters.
- The specified environment size is out of range.

Action: Enter the correct parameters within the specified range and retry the operation.

Invalid file name or file not found

Cause: RENAME or TYPE. You tried to rename a file that was either invalid or not found in the specified directory. TYPE does not allow global file name characters.

Action: Enter the correct file name.

Invalid Function

Cause: COMMANDS. Caused by an application error, this message states that the DOS function is not supported.

Action: Exit the application program.

Invalid keyboard code specified

Cause: KEYB. The keyboard code specified on the command line is not valid.

Action: Correct and retry the operation.

Invalid keyboard ID specified

Cause: KEYB. The keyboard ID specified on the command line is not valid.

Action: Correct and retry the operation.

Invalid keyword

Cause: COMMANDS. You have entered an invalid keyword; for example, entering DELA= instead of DELAY= for the MODE command.

Action: Check your typing and correct it if necessary. If your typing is correct, check the syntax for the appropriate keyword. Retry your request.

Invalid macro definition

Cause: DOSKEY. Your entry contains a syntax error.

Action: Type the following at the command prompt, to get help for the correct syntax:

```
help doskey
```

Invalid media type

Cause: COMMANDS. You have tried to access a disk that is not formatted with DOS, that is only partially formatted, or is damaged.

Action: Format the disk again, or specify another disk to use.

Invalid media or Track 0 bad – disk unusable

Cause: FORMAT. FORMAT was unable to format track 0 on the specified media. This error occurs if:

- Track 0 is unusable. Track 0 is where the boot record, file allocation table, and directory must reside. If track 0 is bad, the disk is unusable.
- The diskette type and drive type are incompatible. You tried to format a double-sided, 320/360KB diskette in a high-capacity, 1.2MB drive; or a high-capacity, 1.2MB diskette in a double-sided, 320/360KB drive.

Action: For the first case, get another diskette and retry the FORMAT command. For the second case, retry the FORMAT command specifying the /4 parameter.

Invalid number of file/directory entries

Cause: FASTOPEN. The FASTOPEN program can work with between 10 and 999 files at one time. You specified a number that is not within that range.

Action: Retype the command using a value between 10 and 999.

Invalid number of parameters

Cause: COMMANDS. You have specified too few or too many parameters for the command you issued.

Action: Check the syntax of the command and re-enter the command.

Invalid option specification

Cause: PCMCIA. The option specified with the command is not valid.

Action: Specify the correct command line parameter; then, try the command again. For information about the PCMCIA commands, check in the "Command Reference" part of this book.

Invalid parameter - *x*

Cause: DOS and COMMANDS. One or more of the parameters entered for these commands are not valid or have been placed in the wrong order.

Action: If the program expects a drive specifier, enter a colon (:) following the drive letter. In other cases, make sure that the character following the slash (/) is valid for the program being run. For JOIN and SUBST, see the command formats in the "Command Reference" part of this book for valid parameters.

Invalid parameter combination

Cause: COMMANDS and REPLACE. One or more of the parameters entered are not valid when entered in this combination.

Action: Review the syntax for this command and enter the command again.

Invalid parameters

Cause: MODE. Parameters are unrecognizable or are in the wrong order.

- The first parameter was other than 40, 80, BW40, BW80, CO40, CO80, MONO, L, or R.
- The adapter that the parameter refers to is not present in the system.

Action: Check the preceding list and correct the command.

Invalid partition table

Cause: FDISK. While attempting to start DOS from your hard disk, the startup procedures detected invalid information in the partition information of the disk.

Action:

1. Start DOS from the hard disk.
2. Use the FDISK command to examine and correct the hard disk partition information.

Invalid path
Invalid Path, not all directories copied

Cause: COMMANDS. The specified path or file name is not correct.

Action: Retype the command using the correct file name and path.

**Invalid path, not directory
or directory not empty**

Cause: RMDIR. This message is caused by one of the following:

- The specified directory was not removed because one of the names you specified in the path was not a valid directory name.
- The directory you specified still contains entries for files or other subdirectories (except for the . and .. entries).

Action: Try one of the following:

- Correct the invalid directory name in the path.
- Delete any files or remove any subdirectories in the directory.

Invalid path or file name

Cause: COPY and RENAME. You specified a directory or file name that does not exist.

Action: Use the correct name. Retry the command after checking for the following:

- Correct spelling of names
- Valid directory names
- Existence of the file in the subdirectory specified

Invalid path or file not found

Cause: ATTRIB. You specified a directory or file name that does not exist.

Action: Use the correct name. Retry the command after checking for the following:

- Correct spelling of names.
- Valid directory names.
- Existence of the file in the subdirectory specified.

Invalid path or System files not found

Cause: COMMANDS. The system files could not be found.

Action: Make sure you specified the correct path to the IBMDOS.COM, IBMBIO.COM, and COMMAND.COM files, and that the files exist on the disk.

Invalid profile statement on line *x*

Cause: GRAPHICS. You have typed an invalid word in the profile statement on the given line.

Action: Review and correct the profile statement.

Invalid socket

Cause: DRVLOCK. The specified socket number is not valid.

Action: Specify a valid socket number and try the command again.

Invalid STACK parameters

Cause: IBMBIO. Invalid combination of either the number of stacks or the stack size specified.

Action: Correct the STACKS command in CONFIG.SYS and restart.

Invalid subdirectory entry

Cause: CHKDSK. Invalid information was detected in the subdirectory whose name precedes this message.

Action: CHKDSK attempts to correct the error if you have used the /F parameter. For more specific information about the error, run CHKDSK with the /V parameter.

Invalid switch - *x*

Cause: DOS and COMMANDS. One or more of the switches entered for the command is not valid, is in the wrong order, or is duplicated. This message might also occur if the command does not use a switch.

Action: Make sure that the character following the slash (/) is valid for the program being run.

Invalid switch type

Cause: FASTOPEN. One or more of the switches entered for the FASTOPEN command is not valid, is in the wrong order, or is duplicated.

Action: Make sure that the character following the slash (/) is valid for the FASTOPEN program being run.

Invalid syntax on DISPLAY.SYS code page driver

Cause: DISPLAY.SYS. The syntax of the DEVICE=DISPLAY.SYS command in the CONFIG.SYS file is incorrect.

Action: Make sure that the parameters are correct, edit the CONFIG.SYS file and restart DOS.

Invalid syntax on PRINTER.SYS code page driver

Cause: PRINTER.SYS. The syntax of the DEVICE=PRINTER.SYS command in the CONFIG.SYS file is incorrect.

Action: Make sure that the parameters are correct, edit the CONFIG.SYS file, and restart DOS.

Invalid time

Cause: COMMANDS. The time format used was not correct for the country code setting in your CONFIG.SYS file.

Action: Type the time using the correct format for the country code setting specified in your CONFIG.SYS file, as indicated by the screen prompt.

Invalid Volume ID

Cause: FORMAT. The volume label entered does not match the volume label on the disk to be formatted.

Action: Issue the VOL command to determine the correct volume label and try again.

Is cross linked on allocation unit *xx*

Cause: CHKDSK. This message appears twice for each cross-linked cluster number, naming the two files in error. The same data block is allocated to both files. Note that some copy-protected software deliberately cross-links its own files and no action should be taken for such files.

Action: No corrective action is taken automatically. Correct the problem by proceeding with the following:

1. Make copies of both files (the COPY command).
2. Delete the original files (the ERASE command).
3. Review the files for validity and edit as necessary.

K

KEYB has not been installed

Cause: KEYB. The KEYB query function was requested before installing KEYB.

Action: Refer to the KEYB command and install.

Keyboard ID specified is inconsistent with the selected keyboard layout

Cause: KEYB. You entered a valid keyboard ID code but it is not compatible with the country.

Action: Refer to the KEYB command in the "Command Reference" part of this book for the keyboard ID code for your particular country.

L

Label not found

Cause: Batch file. A GOTO command named a label that does not exist in the batch file. This caused the system to read to the end of the batch file, ending batch processing.

Action: If you do not want the GOTO to leave the batch file, edit the batch file and put the label in the desired location.

Line too long

Cause: EDLIN. On replacing a string, the replacement caused the line to expand beyond the 253-character limit. The REPLACE text command is ended abnormally.

Action: Split the long line into shorter lines; then issue the REPLACE text command again.

Line to long to join

Cause: E. You tried to join two lines that exceeded the maximum line length of 255 characters.

Action: Break the lines at the maximum margin width and then try rejoining them again.

Lines truncated

Cause: E. You opened a file whose characters exceeded the set margins width. The maximum setting for width (including blank space) is 255 characters.

Action: Do one or more of the following:

- Determine the current margins setting by typing **ma** at the E Editor command line.
- Specify a wider margins setting (if available). For example, type:

  ```
  s margins 1 254
  ```

- Or, exit the file without saving it to maintain file integrity.

List output is not assigned to a device

Cause: PRINT. The list device specified is not a valid print device or no **/d** switch was specified.

Action: Enter PRINT again and specify a valid list device.

LOADHIGH: Invalid argument

Cause: LOADHIGH. The information you provided for a **/l** switch is invalid.

Action: Review the LOADHIGH command. Correct the parameters and re-enter the command. To review LOADHIGH, type HELP LOADHIGH.

LOADHIGH: Invalid filename

Cause: COMMANDS. The format of the specified program is invalid, or the program is not in the path or current directory. The program must be an .EXE or .COM program to run in upper memory.

Action: Make sure you type the file name correctly, that the program exists, and that it is in the path.

Lock violation

Cause: COMMANDS. A program tried to access a file that another program was using.

Action: Close the file being used by the other program, and try program again.

LPT#: not rerouted

Cause: MODE. You canceled the rerouting procedure or the MODE command could not reroute the parallel printer port.

Action: Make sure you have specified the correct parameters with the MODE command.

Type the following at the command prompt, to get help for the correct syntax:

```
help mode
```

M

Maximum number of Logical DOS Drives installed

Cause: FDISK. DOS supports only drive letters A through Z. You have created the maximum number of logical drives (24).

Action: If you need to define a new logical drive, you must first delete an existing logical drive to free a drive letter.

Memory allocation error
Cannot load COMMAND, system halted

Cause: DOS. A program destroyed the area in which DOS keeps track of available memory or, there is no longer enough memory to load the command processor.

Action: Restart DOS on your system. Reducing the number of buffers, device drivers, and resident programs on your system might alleviate this error.

Memory control blocks destroyed

Cause: COMMANDS. IBM DOS detected a memory problem.

Action: Restart your computer.

Missing operating system

Cause: FDISK. When you tried to start DOS from a hard disk, the startup procedures determined that the DOS partition was marked "startable," but that the disk does not contain a copy of DOS.

Action: Start DOS from a diskette and use FORMAT with the /S parameter to place a copy of DOS on the hard disk. Back up your files before doing the FORMAT or they will be lost.

Must enter both /T and /N parameters

Cause: FORMAT. You have entered either the **/t** or **/n** switch by itself.

Action: Because these parameters must be entered together, re-enter the correct format using both parameters.

Must specify COM1, COM2, COM3 or COM4

Cause: MODE. You incorrectly entered MODE option 4.

Action: Retry using correct option.

Must specify destination line number

Cause: EDLIN. A MOVE or COPY command was entered without a destination line number.

Action: Re-enter the command with a valid destination line number.

N

NLSFUNC not installed

Cause: CHCP. NLSFUNC must be installed before running the CHCP command.

Action: Start NLSFUNC. Prepare devices using the MODE command. Retry the CHCP command.

No Append

Cause: APPEND. No directories are currently being searched by APPEND requests. APPEND was previously invoked with only ";" as the append path.

Action: No action is required.

No code page has been selected

Cause: MODE. In the MODE STATUS code page operation, the device reports that no code page is selected for that device.

Action: No action is required. If desired, use MODE to select a code page from the list of prepared code pages that immediately follow this message.

No Drive Specified

Cause: DRIVER.SYS. You did not specify the physical drive number in the DEVICE statement in the CONFIG.SYS file.

Action: See DRIVER.SYS in Chapter 5, "Working With Device Drivers" on page 245, for the correct format to use.

No Extended DOS Partition to delete.

Cause: FDISK. You selected the `Delete extended DOS partition` option, but there is no extended DOS partition on the current hard disk.

Action: Return to the FDISK Options menu and select `Display Partition Data` to review.

No files added

Cause: REPLACE. The /A parameter was specified but all the files on the source already exist in the target directory.

Action: No action is required.

No files found - *x*

Cause: REPLACE. No files described by the source path and file name were found on the source.

Action: No action is required.

No files replaced

Cause: REPLACE. All files found on the target were not replaced because you answered N to all `REPLACE` *<filename>* prompts, or no source files existed on the target.

Action: No action is required.

No hard disks present

Cause: FDISK. The FDISK program was run on a personal computer that:

- Does not have a hard disk, or
- Has a hard disk in the expansion unit and the expansion unit is not powered on, or
- Has a hard disk that is not properly installed.

Action: From the preceding list, determine what caused the problem and act appropriately. Make sure that the expansion unit is powered on first.

No free file handles

Cause: DOS. An attempt to load the command processor failed because there are too many files open.

Action: Increase the number in the FILES command in the configuration file (CONFIG.SYS) and restart DOS.

No logical drives defined

Cause: FDISK. No logical drives have been defined in the extended DOS partition.

Action: To use the space on the disk reserved for the extended DOS partition, create one or more logical DOS drives using the FDISK and format the logical drives using the FORMAT command.

No Logical DOS Drives to delete.

Cause: FDISK. Information message. There are no logical DOS drives in this partition.

Action: Press Esc to return to the previous screen.

No media in drive

Cause: EJECT. You issued the EJECT command on a drive containing no medium.

Action: No action is required.

No Non-DOS Partition to delete

Cause: FDISK. There are no non-DOS partitions on your disk. To see how your disk is currently partitioned, choose option 4, `Display partition information`, from the FDISK main menu.

Action: No action is required.

No partitions defined

Cause: FDISK. There are no partitions defined on the hard disk at the time.

Action: If you wish to use the hard disk for a DOS disk, create a primary DOS partition using FDISK, and format the created drive letter using the FORMAT command.

No partitions to delete

Cause: FDISK. You selected the `Delete partition or or logical DOS drive` option and there are no partitions created on this hard disk.

Action: Return to the FDISK Options menu and select a different option.

No partitions to make active

Cause: FDISK. You selected the `Set active partition` option, but there were no partitions on the current hard disk to be made active.

Action: Return to the FDISK Options menu and select the `Create DOS partition or Logical DOS Drive` option to create a partition, then the `Set active partition` option to make it the active partition.

No Path

Cause: PATH. You typed PATH and pressed Enter to find out what your search path is, but an alternate path for DOS to search for commands and batch files was not specified.

Action: This is an information message. Defining a set of paths is optional. If you want to define the path, enter PATH and the set of paths you want; then press Enter.

No Primary DOS Partition to delete.

Cause: FDISK. You selected the `Delete Primary DOS Partition` option, but there is no DOS partition on the current hard disk.

Action: Return to the FDISK Options menu and select `Display Partition Information` to review.

No room for system on destination disk

Cause: SYS. The destination disk does not contain the required reserved space for DOS; therefore, the system cannot be transferred.

Action: Format a blank diskette (use the FORMAT /S command) and copy any other files to the new diskette.

No room in directory for file

Cause: EDLIN. The directory on the specified disk is full. Your editing changes are lost.

Action: Make sure that your disk has available directory entries and run EDLIN again.

No source drive specified

Cause: COMMANDS. You did not specify a source drive.

Action: Try the command again, specifying both the source and destination drives.

No space to create a DOS partition

Cause: FDISK. You selected the `Create DOS Partition or Logical DOS Drive` option on the current hard disk. There is no space on the current hard disk to create a DOS partition.

Action: Remove or reduce the size of the existing partition. Run FDISK again to create the DOS partition.

No system on default drive

Cause: SYS. The system files to be transferred were not found on the disk or the disk in the default drive.

Action: Select a disk or a disk with the system files on it (DOS disk) and try again.

No tape drive detected

Cause: Backup was unable to find a tape drive.

Action: Try running the Define Equipment command from the Configure menu. After reconfiguring, use the Save as Default command from the File menu to save the new information.

No target drive specified

Cause: COMMANDS. You did not specify a target (destination) source drive.

Action: Try the command again, specifying both the source and destination drives.

Non-DOS disk

Cause: No entry exists for IBMBIO.COM or IBMDOS.COM in the directory; or a disk read error occurred when you started up the system.

Action: Insert a DOS diskette in drive A and restart your system.

Nonremovable drive full

Cause: There is no more space on the destination drive during restoring or backing up.

Action: You should cancel the process and check your destination drive for files you can remove, or check whether the specified drive and path (restore to) is valid. This error may occur if you are backing up to a fixed drive and path.

Non-System disk or disk error
Replace and press any key when ready

Cause: No entry exists for IBMBIO.COM or IBMDOS.COM in the directory; or a disk read error occurred when you started up the system.

Action: Insert a DOS diskette in drive A and restart your system.

***** Not able to restore file *****

Cause: RESTORE. The file you want to restore cannot be opened because of a sharing conflict.

Action: Try again at a later time.

Not able to SYS to name file system

Cause: SYS. You tried to transfer system files to a non-IBM DOS drive.

Action: Specify an IBM DOS drive as the destination drive.

Not enough memory

Cause: E. There is not enough memory for processing the save operation.

Action: You need to increase the amount of available memory in your system. Before you work with your system memory, you might try saving you file by block copying the entire file to a diskette.

To increase your system memory, do one or more of the following:

- Change the BUFFERS parameter in the CONFIG.SYS file to a smaller value.
- Remove device driver programs.

Restart the system and try the command again. If the message still appears, your system does not have enough memory to process the command.

Not enough memory to load help

Cause: IBMBIO. There is not enough memory to load the help.

Action: You need to increase the amount of available memory in your system. You can do this in several ways including:

- Change the BUFFERS parameter in the CONFIG.SYS file to a smaller value.
- Remove device driver programs.

Restart the system and try the command again. If the message still appears, your system does not have enough memory installed.

Not enough room to merge the entire file

Cause: EDLIN. A transfer command was unable to merge the complete contents of the specified file because of insufficient memory. Only part of the file was merged.

Action: Either reduce the size of one of the files being merged, or install more memory.

Not found

Cause: EDLIN. EDLIN could not find the string specified by the REPLACE text or SEARCH text commands within the specified range of lines. Or, if a search is resumed by replying with an N to the `OK?` prompt, no further occurrences of the string were found.

Action: Make sure that you properly use uppercase and lowercase letters for the string to be searched.

Not ready

Cause: COMMANDS. An error occurred while trying to read or write to the indicated drive. Common causes of this error are as follows:

- The diskette drive door is not closed.
- The diskette is not properly formatted.
- The printer is not online.
- The printer does not have enough paper or is busy.

Action: Correct the error and try again. If your default drive is a diskette drive, try typing F to return to the command prompt.

Not resetting hidden file *filename*
Cause: COMMANDS. The specified files with a hidden attribute are not being modified.

Action: If you want to reset attributes on these files, use the ATTRIBUTE command with the -H parameter to remove the hidden attribute from these files before continuing the process.

Not resetting system file *filename*
Cause: COMMANDS. The specified files with a system attribute are not being modified.

Action: If you want to reset attributes on these files, use the ATTRIBUTE command with the -S parameter to remove the system attribute from these files before continuing the process.

O

One or more CON code pages invalid for given keyboard code
Cause: KEYB. You attempted to change your keyboard with a code page combination inconsistent with the CON device code page combinations; the code pages prepared by MODE and the code page supported directly by your hardware. The current CON device code page combination is invalid for the given keyboard code.

Action: Warning message only. The keyboard is loaded, but only the valid code pages are available for the keyboard.

Only non-startable partitions exist
Cause: FDISK. You asked FDISK to change the active partition, but none of the defined partitions can successfully start from the hard disk.

Action: Create a startable partition, such as the primary DOS partition.

Only partitions on Drive 1 can be made active.
Cause: FDISK. The system can only start from the first hard disk, so there is no reason to mark a partition on drive 2 as active.

Action: Use FDISK to mark a partition on drive 1 as active to start from the hard disk.

Only PC card drives are allowed!
Cause: PCMCIA. An attempt was made to use the PCFORMAT command on a drive that was not configured by the PCMFDD command (which enables the drive to emulate a PCMCIA socket either as drive A or B).

Action: Use PCFORMAT only on a drive that is first configured by PCMFFD.

Out of environment space
Cause: DOS. This is an information message caused by one of the following:

- DOS was unable to accept the SET command you just issued because it was unable to expand the area that the environment information is kept in. This normally occurs when you try to add to the environment after loading a program that makes itself resident (PRINT, MODE, or GRAPHICS, for example).

- You tried to expand the environment while in a batch file.

Action: Edit the CONFIG.SYS file to include or increase the **e**: switch setting in the SHELL statement, or remove unnecessary strings from the environment. Restart the system.

Out of memory

Cause: COMMANDS. There is not enough memory for the operation to process.

Action: You need to increase the amount of available memory in your system. To increase your system memory, do one or more of the following:

- Change the BUFFERS parameter in the CONFIG.SYS file to a smaller value.
- Remove device driver programs.

Restart the system and try the command again. If the message still appears, your system does not have enough memory to process the command.

P

Packed file corrupt

Cause: You attempted to load the program into low memory.

Action: Try loading the program using LOADFIX.

Note: LOADFIX loads a program above the first 64K of memory and runs the the program.

Parameter format not correct

Cause: DOS and COMMANDS. One or more of the parameters entered for the command is not valid, is in the wrong order, or is duplicated.

Action: Make sure that the character following the slash (/) is valid for the program being run, and that the command syntax is correct.

Parameter value not allowed

Cause: DOS and COMMANDS. One or more of the parameters entered for the command is not valid.

Action: Make sure that the value following the slash (/) is valid for the program being run.

Parameter value not in allowed range

Cause: DOS and COMMANDS. One or more of the parameter values entered is too high or too low.

Action: Check your typing to see if you misspelled a parameter and that the parameter is within the valid range.

Parameters not compatible

Cause: FORMAT. You attempted to use two parameters that are not compatible with each other (/**b** and /**v**, for example); the parameters specified do not apply to the drive or media.

Action: Review the FORMAT command. Correct the parameters and re-enter the command.

Parameters not compatible with hard disk

Cause: FORMAT. You have used a switch that is not compatible with the specified drive.

Action: Review the FORMAT command. Correct the parameters and re-enter the command.

Parameters not supported

Cause: FORMAT. You tried to format a disk using the **/n** and **/t** separately.

Action: **/n** and **/t** must be used together. Try the command again, using the correct format.

Parameters not supported by drive

Cause: FORMAT. This message occurs during initialization. An invalid parameter that cannot be supported on a specific type of hardware was entered.

Action: Review the FORMAT command parameters for further information.

Parse error #

Cause: DOS. An error occurred on the command line.

Action: The "#" position in this message represents a number that corresponds to a particular error that caused this message to be displayed.

The specific causes and actions for these errors are listed in this book in alphabetic order. Use the following list to find the message that corresponds to the number displayed on your screen:

1. - Too many parameters
2. - Required parameter missing
3. - Invalid switch
4. - Invalid keyword
6. - Parameter value not in allowed range
7. - Parameter value not allowed
8. - Parameter value not allowed
9. - Parameter format not correct
10. - Invalid parameter
11. - Invalid parameter combination

Partition selected is not a Non-DOS Partition

Cause: FDISK. You selected the option to delete a non-DOS partition; however, the partition you selected is not a non-DOS partition.

Action: Select a non-DOS partition, or press ESC to return to the FDISK main menu.

Partition selected is not a Primary DOS Partition

Cause: FDISK. You tried to delete a non-DOS primary partition; however, this option can be used only for DOS primary partitions.

Action: Specify DOS primary partition, or press ESC to return to the FDISK main menu.

**Partition selected (#) is not startable,
active partition not changed.**

Cause: FDISK. The partition that you selected to be marked active is not a startable partition. The active partition was not changed.

Action: Select another partition to be marked active.

Path not found

Cause: DOS and COMMANDS. A file or path named in a command or command parameter does not exist in the directory of the specified (or default) drive.

Action: Retry the command, using the correct path and file name.

Path too long

Cause: COMMANDS. The path specified as a parameter was greater than 63 characters.

Action: Correct the path and try again.

Path name too long

Cause: COMMANDS. The path for the file you specified is longer than 63 characters.

Action: You might have to move to the directory containing the file you want to process before using the current command.

Primary DOS partition already exists.

Cause: FDISK. A primary partition for DOS has already been set up.

Action: Return to the FDISK Options menu and select a different option.

PRINT queue is full

Cause: PRINT. You tried to add more than the limit of 10 files to the print queue; 10 files is the default. You can set the limit to 32 files. See the PRINT command.

Action: Wait until a file is printed before you add another file to the print queue.

Printbox ID not in GRAPHICS profile

Cause: GRAPHICS. At the command prompt, you specified a PRINTBOX ID that was not defined in the GRAPHICS profile.

Action: The ID specifies a print box size and should match the first operand of a PRINTBOX statement in the GRAPHICS profile.

Printer error

Cause: MODE. The MODE command (option 1) was unable to set the printer mode because:

- An I/O error occurred.
- The printer is out of paper (or switch is off).
- The printer timed out (is not ready).
- The printer is off-line.

Action: Determine which of the previous conditions caused the error message and correct it.

Printer out of paper error

Cause: PRINT. The printer is out of paper or not turned on.

Action: Turn the printer on, press the "Online" switch, or add paper and retry.

Printer type not in GRAPHICS profile

Cause: GRAPHICS. At the command line, you specified a certain printer type that is not defined in the GRAPHICS profile.

Action: See the parameters for the GRAPHICS command in this book on page 120 for further information about the valid printer types.

Problem with read

Cause: COMMANDS. There are bad sectors within the directory or file allocation table (FAT) structure. An error occurred when accessing the directory.

Action: If you are trying to format your hard disk, try formatting the disk to make it usable again.

Problem with write

Cause: COMMANDS. There are bad sectors within the directory or file allocation table (FAT) structure. This is a write problem.

Action: Try formatting the disk to make it usable again.

Profile statement out of sequence on line *x*

Cause: GRAPHICS. A profile statement is out of the correct sequence on line *x*.

Action: Review and correct the sequence of the statement.

Program too big to fit in memory

Cause: DOS. The file containing the external command cannot be loaded because it is larger than the available free memory.

Action: Reduce the number in the BUFFERS parameter in your CONFIG.SYS file (if you have specified BUFFERS), restart your system, and reissue the command.

If the message reappears, your system does not have enough memory to process the command.

Processing cannot continue

Cause: CHKDSK. Your computer might have run out of memory, or the file allocation table (FAT) might be damaged.

Action: If you continue to see this message, back up the information and reformat the disk.

R

RAMDrive:Error in extended memory allocation

Cause: RAMDRIVE.SYS. The extended memory manager reported an error to RAMDRIVE.SYS while attempting to allocate memory for the RAM drive.

Action: Run memory diagnostic software on your system. Consult the documentation for your extended memory manager.

RAMDrive:Expanded Memory Manager not present

Cause: RAMDRIVE.SYS. You included the /**a** switch in the RAMDRIVE.SYS command line. However, you failed to load the expanded memory manager before RAMDRIVE.SYS. Therefore, RAMDRIVE.SYS could not be installed.

Action: You must always load RAMDRIVE.SYS after the expanded memory manager when the /**a** switch is used.

RAMDrive:Expanded Memory Status shows error

Cause: RAMDRIVE.SYS. DOS detected an error in the expanded memory adapter while trying to install RAMDRIVE.SYS. RAMDRIVE.SYS could not be installed.

Action: Consult your expanded memory adapter documentation on memory errors.

RAMDrive:Extended Memory Manager not present

Cause: RAMDRIVE.SYS. RAMDRIVE.SYS could not find the extended memory manager.

Action: Install HIMEM.SYS or another XMS extended memory manager. Make sure the device command that installs the extended memory manager comes before the device command for RAMDRIVE.SYS in the CONFIG.SYS file.

RAMDrive:Insufficient memory

Cause: RAMDRIVE.SYS. There is not enough memory available to set up a RAM drive. RAMDRIVE.SYS was not installed.

Action: Free some extended memory or obtain more memory.

RAMDrive:Invalid parameter

Cause: RAMDRIVE.SYS. The parameters you specified in your CONFIG.SYS entry for RAMDRIVE.SYS are not correct. RAMDRIVE.SYS was not installed.

Action: Check to see if you specified too many parameters, if one of your numeric parameters is not valid, if you specified conflicting switches, or if you specified too many switches.

RAMDrive:I/O error accessing drive memory

Cause: RAMDRIVE.SYS. While setting up the RAMDRIVE.SYS driver, an error was detected in the memory being accessed for the RAM drive. RAMDRIVE.SYS was not installed.

Action: Run the memory test for the memory on which you were attempting to install the RAM drive.

RAMDrive:No extended memory available

Cause: RAMDRIVE.SYS. Your computer has no memory available for RAM drives. RAMDRIVE.SYS was not installed.

Action: Free some extended memory or obtain more memory.

RATE and DELAY must be specified together

Cause: MODE. The typematic rate and delay must both be set at the same time.

Action: Specify both rate and delay values when you want to change the setting of either.

Read error in: *x*:\level 1\level 2.

Cause: EDLIN. An error occurred while reading file *x:\xxxx\xxxx* into memory.

Action: Copy the file or a backup of the file to a different disk and try again.

Read fault error

Cause: COMMANDS. DOS was unable to read the data from the device.

Action: Make sure the device is installed and connected correctly, and that the medium is properly inserted in the drive.

If you get the same message, select Abort and rerun the command with different medium.

Requested logical drive size exceeds the maximum available space.

Cause: FDISK. The cylinder size entered for the logical drive is larger than the maximum available space in the extended DOS partition. FDISK displays the maximum available size for the partition as the default entry value.

Action: Enter a value that is less than or equal to the default size displayed.

Requested partition size exceeds the maximum available space.

Cause: FDISK. The cylinder size entered for the partition size is larger than the maximum available space on the hard disk. FDISK displays the maximum available size for the partition as the default entry value.

Action: Enter a value that is less than or equal to the default size displayed.

Required font not loaded

Cause: MODE. The console device driver (DISPLAY.SYS) does not have the font size required to set the requested number of lines in the current code page.

Action: Set the maximum number of subfonts in the DEVICE=DISPLAY.SYS statement in the CONFIG.SYS file; restart DOS.

Required parameter missing

Cause: COMMANDS. You tried to use a command without using a required parameter.

Action: To find out which parameters are valid, at the command prompt, type `HELP` followed by the command name.

Required profile statement missing before line *x*

Cause: GRAPHICS. A profile statement has been left out of the sequence of statements before line *x*.

Action: Review GRAPHICS statements and add the missing statement before line *x*.

Requires IBM DOS Version 4.0 or later

Cause: CHOICE. You are working with a version of DOS that is earlier than DOS Version 4.0.

Action: Upgrade your version of DOS.

Restore file sequence error

Cause: RESTORE. The file was not restored because the diskettes were not inserted in sequential order.

Action: Retry the restore, inserting the diskettes in sequential order.

S

Same drive specified more than once

Cause: FASTOPEN. The same drive letter was specified more than once.

Action: Re-enter the command, specifying each drive letter only once.

Same parameter entered twice

Cause: FORMAT. You have entered one of the following parameters more than one time: **/t**, **/n**, **/f**, or **/v**.

Action: Re-enter the correct parameter.

SECOND diskette bad or incompatible

Cause: DISKCOMP. You tried to read from the wrong media in an incompatible drive.

Action: Check your diskette by using the CHKDSK command.

Sector not found

Cause: The sector containing the data could not be located on the disk.

Action: Refer to "Responses" on page 303 of this book.

If you get the same message, select Abort and rerun the command with a different disk.

Sector size too large in file

Cause: IBMBIO. The device driver named in <filename> specifies a device sector size larger than the devices previously defined to DOS.

Action: Reduce the sector size to conform with the sector size of DOS.

Seek error

Cause: The hard disk or disk drive was unable to find the proper track on the disk.

Action:

- Make sure that the diskette is properly inserted in the drive.
- Try a different drive.
- Run CHKDSK.

Refer to "Responses" on page 303 in this book.

SHARE cannot be installed under DOSSHELL

Cause: SHARE. SHARE cannot be set up while operating under the DOS Shell.

Action: To set up SHARE, quit the DOS Shell; then, install SHARE from the command prompt, and start DOSSHELL again.

Sharing violation

Cause: COMMANDS. You tried to access a file using a sharing mode not allowed now. Normally, this occurs when someone else is accessing the file compatibility mode for writing, or in a sharing mode that does not allow you to access the file at the same time.

Action: Wait a short time and try again. If you need to respond, select "Retry" as your first choice. If that fails, select "Abort".

SMARTDrive configuration is too large

Cause: SMARTDRV. There is not enough memory for the configuration you specified.

Action: For help, type **help smartdrv** or see command "SMARTDRV" on page 217.

SMARTDrive: Bad extended memory manager control chain

Cause: The XMS control chain was damaged by a program not using XMS. SMARTDRV.SYS is not installed.

Action: Remove device drivers that come before SMARTDRV.SYS in your CONFIG.SYS, one at a time, to determine which is the offending program.

SMARTDrive: Error in extended memory allocation

Cause: SMARTDRV.SYS. The extended memory manager reported an error to SMARTDRV.SYS while attempting to allocate memory for the disk cache. SMARTDRV.SYS is not installed.

Action: Run memory diagnostic software on your system. Consult the documentation for your extended memory manager.

SMARTDrive: Expanded Memory Manager not present

Cause: SMARTDRV.SYS. You included the /a switch in the SMARTDRV.SYS command line but SMARTDRV.SYS could not find the expanded memory manager. SMARTDRV.SYS is not installed.

Action: You must always load SMARTDRV.SYS after the expanded memory manager when the /a switch is used.

SMARTDrive: Extended Memory Manager not present

Cause: SMARTDRV.SYS. SMARTDRV.SYS could not find the extended memory manager. SMARTDRV.SYS is not installed.

Action: Install HIMEM.SYS or another XMS extended memory manager. Make sure the device command that installs the extended memory manager comes before the device command for SMARTDRV.SYS in the CONFIG.SYS file.

SMARTDrive: Expanded Memory Manager Status shows error

Cause: SMARTDRV.SYS. While trying to set up SMARTDRV.SYS in expanded memory, DOS detected an error. DOS will not install the SMARTDRV.SYS driver.

Action: Run the expanded memory diagnostics to check the expanded memory.

SMARTDrive: I/O error accessing cache memory

Cause: SMARTDRV.SYS. DOS detected an error while trying to set up SMARTDRV.SYS. DOS will not install the SMARTDRV.SYS driver.

Action: Run the memory test for the memory on which SMARTDRV.SYS is attempting to use as the cache.

SMARTDrive: Insufficient memory

Cause: SMARTDRV.SYS. Your system has insufficient memory available for SMARTDRV.SYS. DOS will not install the SMARTDRV.SYS driver.

Action: If you want to use the SMARTDRV.SYS driver, you must add memory to your system.

SMARTDrive: Invalid parameter

Cause: SMARTDRV.SYS. The SMARTDRV.SYS command line contains an invalid switch option, an invalid value for memory, or some other invalid entry.

Action: Edit your CONFIG.SYS file and change the incorrect SMARTDRV.SYS line.

SMARTDrive: No extended memory available

Cause: SMARTDRV.SYS. Your system does not have enough memory available for SMARTDRV.SYS. DOS will not install the SMARTDRV.SYS program.

Action: You must add memory to your system or reduce the amount of memory used by other device drivers.

SMARTDrive: No hard drives on system

Cause: SMARTDRV.SYS. Your system has no hard drive. DOS will not install the SMARTDRV.SYS program.

Action: SMARTDRV.SYS only works with hard drives.

SMARTDrive: Too many bytes per track on hard drive

Cause: SMARTDRV.SYS. Your system has a hard disk drive that SMARTDRV.SYS does not understand. DOS will not install the SMARTDRV.SYS program.

Action: SMARTDRV.SYS cannot be used with this hard disk drive.

Socket *x:* does not support DRVLOCK

Cause: DRVLOCK. The specified socket does not support the lock and unlock functions of DRVLOCK.

Action: Refer to the documentation that came with the socket to verify if it supports the DRVLOCK functions.

Socket Services not loaded

Cause: PCMCIA. An attempt was made to install Card Services without first having Socket Services loaded.

Action: Load Socket Services; then, install Card Services. For installation information, see the *User's Guide.*

Socket Services version must be 2.0

Cause: PCMCIA. The Socket Services Version on your system is not 2.0.

Action: Install Socket Services Version 2.0; then, install Card Services Version 2.0.

Source and destination are the same

Cause: MOVE. You have entered the same drive as the source and target.

Action: Specify a different drive letter for the source and target drives.

Source destination conflict

Cause: E. The source and the target drives are the same.

Action: Do not specify the same drive for both the source and the target.

Source and target drives are the same

Cause: RESTORE. You have entered the same drive as the source and target.

Action: Specify a different drive letter for the source and target drives.

SOURCE diskette bad or incompatible

Cause: DISKCOPY. The errors occurred while reading the diskette. The error might have been caused by bad sectors, or the diskette in the source drive might not be compatible with the source drive type (a high-capacity diskette in a 360KB disk drive, for example).

Action: Check your diskette.

Source does not contain backup files

Cause: RESTORE. The source media does not contain files created by the BACKUP command.

Action: No action is required.

Source path and target drive cannot be the same

Cause: COMMANDS. The specified source path and the destination drive are the same.

Action: Specify a path that is different from the destination drive.

Source path required

Cause: COMMANDS. You did not specify a source path.

Action: Specify the path of the original file or directory.

Specify Kilobytes or Megabytes by using M or K

Cause: PCMCIA. You used the PCFORMAT command on a SRAM card without designating a format size (*m* for megabytes or *k* for kilobytes) with the **/s** switch. For example, **/s**:204K or **/s**:2M.

Action: Specify the appropriate format size designator. For more information, see command "PCFORMAT" on page 177.

String not found

Cause: E. You specified a search for a character string that does not exist in the file.

Action: Specify another search string, or try searching for the string in another file that might contain it.

Switch *x* is not valid

Cause: DELTREE. One or more of the switches entered for the command is not valid, is in the wrong order, or is duplicated. This message might also occur if the command does not use a switch.

Action: Make sure that the character following the slash (/) is valid for the DELTREE command; then, try the command again. For information about valid DELTREE command switches, see command "DELTREE" on page 68 or type **deltree /?** at the DOS command line prompt.

Syntax error

Cause: DOS and COMMANDS. The command format you typed is incorrect.

Action: Make sure that you have used the correct format for this command. For information about the command, check in the "Command Reference" part of this book or type at the DOS command line prompt the the name of the command followed by **/?**.

Syntax error in command line!

Cause: PCMCIA. The specified command line parameter is not correct.

Action: Specify the correct command line parameter and try the command again. For information about the command, check in the "Command Reference" part of this book.

Syntax errors in GRAPHICS profile

Cause: GRAPHICS. You have entered an incorrect syntax in your statements in the GRAPHICS.PRO file. For example, words that are in reverse order.

Action: Review the statements and correct the syntax errors.

SYS cannot operate on target drive

Cause: SYS. The target drive you specified is not supported by the SYS command.

Action: Specify a valid target drive.

System resource exhausted

Cause: COMMANDS. The system is unable to obtain all needed resources to process this command. All available storage is currently in use.

Action: Try one of the following:

- Reduce the number of programs presently running on your system.

- Edit the CONFIG.SYS file to reduce the value in the BUFFERS statement, or reduce the size allocated for a virtual disk. Restart the system and retry the command.
- Install additional memory in your system.
- Contact the supplier of the application.

T

Tape drive does not have long-tape capability

Cause: Your drive is unable to handle extended-length tapes.

Action: Use a drive that can handle extended-length tapes.

Tape is not empty

Cause: This error message occurs if you are prompted to insert another tape during a backup.

Action: This tape must be empty. If it is not, you are given the option to erase the tape or insert a new tape.

Tape needs certification

Cause: You attempted to use a tape that is not certified.

Action: You can insert a new tape, or have Backup certify the tape for you.

TARGET diskette bad or incompatible

Cause: DISKCOPY. You tried to write to the wrong media in an incompatible drive.

Action: Check your diskette by using the CHKDSK command.

Target diskette may be unusable

Cause: DISKCOPY. This message follows an unrecoverable read, write, or verify error message. The copy on the target disk may be incomplete because of the unrecoverable I/O error.

Action:

- If an error is on the target disk, use a different disk for your target, and retry the DISKCOPY command.
- If the error is on the source disk, copy all files from the source disk to another disk; then try to reformat the source disk.

Target diskette unusable

Cause: DISKCOPY. Errors met during the copying procedure show that the target disk is dirty, damaged, or of poor quality, or the drive is malfunctioning on some tracks.

Action: Change the target drive or the disk and try again.

Text already marked

Cause: E. You tried to mark a block, word or character text after you previously marked a block, word or character.

Action: Mark one block, word or character at a time; then, try again.

Text not marked

Cause: E. You tried to move or copy text before you marked it.

Action: Mark the text you want to copy or move; then, try again.

The disk might not be formatted

Cause: COMMANDS. The operation was not successful because the disk might be bad, the wrong type, or not formatted.

Action: Determine if the disk is usable. If it is not, use another disk.

The help file *x* has incorrect version information

Cause: HELP. You are using a file whose version does not match the version you initially loaded.

Action: Determine why you are accessing the wrong version (often the result of your PATH setting) and change your setup to find the correct version.

The last file was not restored

Cause: RESTORE. You stopped RESTORE before it completely restored the last file listed, or there was not enough room on the hard disk, and RESTORE deleted the partially restored file.

Action: If RESTORE has ended, you can re-enter the RESTORE command with the file name of the files not restored to continue from the point where RESTORE stopped.

If the problem occurred because you ran out of room on the hard disk, you must evaluate which files to keep and which ones to delete. Continue processing RESTORE.

The master boot code has NOT been updated

Cause: FDISK. There is a physical problem reading or writing to the hard disk. This could indicate a bad sector on the disk where the master (startup) record is located.

Action: Use the FDISK command to delete the partition and create a new one, or use the low-level format that came with the hard disk to format the disk.

The only startable partition on Drive 1 is already set active.

Cause: FDISK. You specified setting your primary partition active. The partition is already active and only one can be set active at a given time.

Action: No action is required.

There is not enough disk space for system files

Cause: FORMAT. The disk has too much data stored on it.

Action: Delete some files from this disk to increase the disk space, or use another disk.

Too many arguments in command line

Cause: MORE. You have specified too many or conflicting parameters with the command.

Action: Make sure the syntax of the command is correct, and that you only specify one process at a time.

For syntax information, type `help` followed by the command name at the command prompt.

Too many block devices

Cause: IBMBIO. You attempted to install more than the system limit of 26 block device units.

Action: Change the DEVICE value in the CONFIG.SYS file so there are only 26 block device units (including those automatically installed by DOS for disk drives). See the *User's Guide* for more information.

Too many code pages specified

Cause: DISPLAY.SYS. The number of code pages you specified to be supported on your display device is not within the range of the valid values. For EGA or VGA displays, the value can be between 0 and 6.

Action: Edit the display device statement in your CONFIG.SYS file to specify the correct value; then, restart your computer. The following is an example of what the display device statement looks like:

```
device=c:\dos\display.sys con:=(ega,,n)
```

The *n* represents the number of code page fonts you want to load or use.

Too many drive entries

Cause: FASTOPEN. Too many drive letters were specified.

Action: Re-enter the command with fewer drive letters.

Too many files

Cause: E. The number files you tried to edit exceeded 39. The editor does not allow more than 39 files in a ring.

Action: Decrease the number of files in the ring.

Too many files open

Cause: EDLIN, COMP, or LABEL. You attempted to open the specified file, but were not able to.

Action: Increase the FILES value in the CONFIG.SYS file.

Too many file/directory entries

Cause: FASTOPEN. You have specified too many directory or file entries in the FASTOPEN statement. The total value of *n*'s must range from 10 through 999.

Action: Review your entries and make sure that the values given are in the range of 10 through 999.

Too many parameters

Cause: COMMANDS. Too many positional parameters were entered on the command line.

Action: Review the command syntax and re-enter the command.

Too many rings

Cause: E. You tried to open to many file rings.

Action: Decrease the number of rings you want to edit.

Too much of memory fragmentation; MEM /C cannot be done

Cause: MEM. The memory is too fragmented, or an application that ran before the MEM /C command has damaged the memory areas.

Note: The MEM command can still be used alone or with the /P parameter.

Action: Restart your computer, or, if you are using IBM DOS through a DOS box (such as a windowing-type program), close the windowing program and try again.

U

Unable to access drive *d*.

Cause: FDISK. FDISK is unable to access the drive you specified.

Action: Turn off your system; then turn it on and retry.

Unable to create destination

Cause: MOVE. The destination you specified was invalid.

Action: Try the command again using a valid destination.

[Unable to create directory]

Cause: DOS and COMMANDS. You might have a directory with the same name, the path might not be correct, the disk might be full or write-protected, or the root directory might be full.

Action: Use the DIR command with the /A parameter to make sure there are no hidden directories on the disk with the same name, and to determine the amount of free space the disk contains. Also, make sure the path is valid. If you tried to create the directory in the root directory and the root directory was full, move or delete some unneeded files to increase disk space, or create the directory on a different drive.

Unable to create KEYB table in resident memory

Cause: KEYB. The KEYB command was previously installed, and allocated a specific amount of resident memory for the tables. The requested configuration exceeds that resident memory.

Action: Restart DOS and reinstall KEYB with a new configuration.

Unable to do requested command

Cause: COMMANDS. The network cannot perform the command because of errors.

Action: Check to make sure the network connection is secure and that the network is installed correctly. If the connection is secure and the installation is correct, you might try restarting your computer. If you continue to see this message, contact your network administrator, or reinstall the network.

Unable to initialize Card Services. Error Code:1

Cause: When attempting to load Card Services.

Action: Refer to the README.TXT file in your DOS directory for information about Socket Services.

Unable to open file

Cause: Backup has encountered a file that is open and in use.

Action: You can set the Retry on Busy File option (from the Configure menu) to keep trying for a set period of time. However, the default is to skip open or busy files and continue.

Unable to perform REFRESH operation

Cause: MODE. The printer device driver (PRINTER.SYS) does not have a copy of the code page in its Random Access Memory (RAM) to download to the specified printer.

Action: Prepare and select the desired code page. To allow refreshes in the future, make sure that the number of buffers ("*n*" in DEVICE=PRINTER.SYS) is 1 or greater.

Unable to read media

Cause: The disk might not be inserted properly.

Action: Check that the disk is properly inserted and the drive latch is secure.

Unable to read tape header

Cause: A tape header is similar to a directory in a disk. It is a map to the contents of the tape. If it somehow gets damaged or erased, the information stored on the tape is lost.

Action:

Unable to reload with profile supplied

Cause: GRAPHICS. You already loaded GRAPHICS and want to load it a second time specifying a different profile. The memory allocated on the first load is too small.

Action: Restart your system.

Unable to write BOOT

Cause: FORMAT. The first track of the diskette or DOS partition is bad. The BOOT record could not be written on it. The diskette or DOS partition is not usable.

Action: Get another diskette and retry the FORMAT command.

Unable to write to destination

Cause: MOVE. There is not enough space at the specified destination.

Action: Do one or more of the following:

- Make sure there is enough space at the specified destination.
- Erase unneeded files from the specified destination.
- Specify a different destination.

Then, try the operation again.

Unknown Media type

Cause: QCONFIG.SYS. You have tried to access media that is not formatted with DOS, that is only partially formatted, that is damaged, or that is incompatible.

Action: Format the disk, specify another disk to use, or make sure that the media is compatible.

Unrecognized command in CONFIG.SYS

Cause: IBMBIO. An invalid command was detected in the configuration file CONFIG.SYS.

Action: Edit the file, correct the invalid command, and restart DOS.

Unrecoverable error in directory

Cause: CHKDSK. CHKDSK met an error while checking the directory.

Action: No action is required.

Unrecoverable read error on drive *d*, side *n*, track *xx*

Cause: DISKCOMP or DISKCOPY. Four attempts were made to read the data from the diskette in the specified drive. The data could not be read from the indicated track and side.

Action: If the error occurred on the target diskette (just created by DISKCOPY), get a different diskette and retry the DISKCOPY and DISKCOMP commands. Otherwise, copy all files from the damaged diskette to another diskette. Reformat the bad diskette or discard it.

Unrecoverable write error on drive *d*
side *n*, track *xx*

Cause: DISKCOPY. Several attempts were made to write the data to the target diskette. DISKCOPY continues copying, but the copy may contain incomplete data.

Action: Get a different diskette and re-enter the DISKCOPY command. Use FORMAT on the bad diskette to see if it can be reused. If it is a bad diskette, discard it.

Unrecognized switch

Cause: DOS and COMMANDS. One or more of the switches entered for the command is not valid, is in the wrong order, or is duplicated. This message might also occur if the command does not use a switch.

Action: Make sure that the character following the slash (/) is valid for the program being run. For information about the command, check in the "Command Reference" part of this book or type at the DOS command line prompt the name of the command followed by **/?**.

Unsupported tape type

Cause: Backup does not support the tape you are trying to use.

Action: Make sure it is a DC2000, DC2080, or DC2120 1/4+ tape.

V

Volume label does not match.

Cause: FDISK. The volume label you entered is different from the label assigned to this drive.

Action: Enter the correct volume label.

Volume label is not supported with /8 parameter

Cause: FORMAT. You tried to used the LABEL command on a disk formatted with the /8 parameter. A volume label cannot be specified for a disk formatted with this parameter.

Action: No action is required.

W

WARNING - directory full

Cause: RECOVER. There is insufficient directory space to recover more files.

Action: Copy some of the files to another diskette, erase them from this diskette, and run RECOVER again.

WARNING! No files were found to restore

Cause: RESTORE. The specified files could not be found on the backup disk.

Action: Make sure you typed the file names correctly, and that the files exist.

WARNING! No partitions are set active - disk 1 is not startable unless a partition is set active

Cause: FDISK. Your system does not have an active primary partition. You cannot start your computer from the hard disk without an active primary partition.

Action: To set an active partition, choose option 2, **Set active partition**, from the FDISK main menu.

WARNING! The partition set as active is not startable

Cause: FDISK. The partition set active does not contain the system files.

Action: Make sure the IBMDOS.COM, IBMBIO.COM, and COMMAND.COM files are present in the primary DOS partition.

WARNING! Unable to use a disk cache on the specified drive

Cause: SMARTDRV. Disk cache is not supported on the specified drive or an invalid drive was specified.

Action: Specify a valid drive.

Write failure, diskette unusable

Cause: COMMANDS. The destination disk cannot be used.

Action: Specify a different destination disk.

Write fault error

Cause: COMMANDS. You specified a device that is not assigned in your system, or the device is not connected properly.

Action: Make sure that the device is properly connected and installed in your system, and that you typed the command with the correct device specification.

Write protect error writing drive *x*:

Cause: An attempt was made to write to a write-protected diskette.

Action: Investigate carefully before you decide to write to a write-protected diskette.

Your default choice is not one of the valid choices

Cause: CHOICE. The choice you used as the default in your batch program is not valid. For information about valid choice selections, see command “CHOICE” on page 37 or type **choice /?** at the DOS command line prompt.

Action: Specify a valid choice in the batch program.

Chapter 8. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights or other legally protectable rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*) in this publication, are trademarks or service marks of the IBM Corporation in the United States or other countries:

IBM
DisplayWrite
PS/2
OS/2
Proprinter
Quietwriter
PC/XT
Personal System/2
1420 Micro Channel

The following terms, denoted by a double asterisk (**) in this publication, are trademarks of other companies as follows:

PostScript	Adobe Systems Incorporated
AST	AST Research, Inc.
QuattroPro	Borland
Delete Sentry	Central Point Software
Handwriter	Communication Intelligence Corporation
PenDOS	Communication Intelligence Corporation
Everyday DOS	Computer Tyme
CSS	CSS Laboratories, Inc.
Dell	Dell Corporation

DeskJet	Hewlett-Packard Company
Hercules	Hercules Computer Technology
Hewlett-Packard	Hewlett-Packard Company
HP	Hewlett-Packard Company
LaserJet	Hewlett-Packard Company
PaintJet	Hewlett-Packard Company
Intel	Intel Corporation
Intel Series I and II Flash Cards	Intel Corporation
Lotus	Lotus Development Corporation
1-2-3	Lotus Development Corporation
Microsoft	Microsoft Corporation
Microsoft OAK	Microsoft Corporation
Microsoft Flash File System I and II	Microsoft Corporation
MS	Microsoft Corporation
MS-DOS	Microsoft Corporation
Windows	Microsoft Corporation
PCMCIA	Personal Computer Memory Card International Association
Philips	Philips Consumer Electronics Company
Phoenix	Phoenix Technologies, Ltd.
Phoenix Card Manager Plus	Phoenix Technologies, Ltd.
Windows	Microsoft Corporation
Epson	Seiko Epson Kabushiki Kaisha
Toshiba	Toshiba Corporation
Video 7	Video Seven, Inc.

Index

0 87944 06620 0

S71G-3634-00

71G3634

P71G3634